2013

Protein Identification Using Bayesian Stochastic Search

Christina Nicole Lewis
University of South Carolina - Columbia

PROTEIN IDENTIFICATION USING BAYESIAN STOCHASTIC SEARCH

by

Christina Nicole Lewis

Bachelor of Science
East Tennessee State University 2006
Master of Statistics
University of Tennessee 2008

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Statistics

College of Arts and Sciences

University of South Carolina

2013

Accepted by:

David Hitchcock, Major Professor

Ian Dryden, Committee Member

Xiaoyan Lin, Committee Member

John R. Rose, External Examiner

Lacy Ford, Vice Provost and Dean of Graduate Studies

DEDICATION

To all those who supported, inspired, and most importantly, believed in me

ACKNOWLEDGMENTS

I would first like to acknowledge both of my Ph.D. advisors Dr. David Hitchcock and Dr. Ian Dryden whom I am forever grateful for the guidance and lessons they have bestowed upon me. Both have inspired me to never stop learning. They are true definition of a mentor providing guidance, insight, and wisdom. Without their advice, encouragement, and support, I would not be were I am today.

I also wish to acknowledge Dr. John Rose. I want to thank him for his contributions to my research. His continuing research in the field of proteomics sparked an interest me that flourished immensely. He and his graduate student Jimmy Cleveland provided me with the knowledge and background to succeed in my research. Next I wish to acknowledge Dr. Xiaoyan Lin for her knowledge and directrion, which enriched my research agenda.

I also with to acknowledge the Department of Statistics at the University of South Carolina and all the professors in the department. I want to thank them for the extensive and in depth understanding of the field of statistics. Their efforts have strengthened my academic and research paths. I greatly appreciate all the opportunities the department has provided me.

Finally, I wish to thank my family for their unconditional love and support. To my husband who is truly my best friend, without you, none of this would have been possible. You have been rock through this entire process and your positivity encouraged me to push forward. To my mom, you have always been my hero. You are a strong, confident, independent woman that I can only aspire to be. Thank you for being my mom and for your continued support in all the endeavors' I pursue. To papaw, if were not for you, I would not be where I am today and I am forever indebted. I will always remember your words:

"Remember that train Nicole. I think I can. I think I can." I have and will continue to seek your guidance and wisdom. To my younger brother, words cannot express the role you have played in my life. Continue to always set your goals high because I know you will achieve them. Finally, to all of my family and friends, I thank you for always believing in me even when I did not. I am eternally grateful for your words of advice and inspiration.

ABSTRACT

Current methods for protein identification in tandem mass spectrometry (MS/MS) involve database searches or de novo peptide sequencing, with database searches being the standard method. With database searches, issues arise when the species is not in the database. Shortcomings of de novo peptide sequencing and database searches include chemical noise, overly complex fragments, and incomplete b and y ion sequences. Here we present a Bayesian approach to identifying peptides. Our model uses prior information about the average relative abundances of bond cleavages and the prior probability of any particular amino acid sequence. The proposed likelihood function is composed of two overall distance measures, which measure how close an observed spectrum is to a theoretical scan for a peptide. A Markov chain Monte Carlo (MCMC) algorithm is employed to simulate candidate choices from the posterior distribution of the peptide sequence. The true peptide is estimated as the peptide with the largest posterior density. In addition, our method is designed to rank top candidate peptides according to their approximate posterior densities, which allows one to see the relative uncertainty in the "best" choice. A simulation study was carried out to ensure our algorithm is performing accurately. Two different noise structures were explored: a Laplace noise structure and a Poisson noise structure. Simulation studies showed our methods are promising. Our motivating data come from the Pacific Northwest National Laboratory (PNNL) and the dataset is from the salmonella typhimurium species. The dataset is a set of doubly charged tryptic peptides. When our method was applied to peptides from this dataset, the true peptide was captured among the list of the top estimated peptides.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Proteomics is a vast analysis of proteins, particularly their structure, function, abundances, and variations and modifications. In proteomics, scientists begin with the protein and work backwards to determine the gene that is responsible for its production. Arguments can be made for studying proteins as opposed to mRNA or DNA. Proteins are constantly changing and vary with health or disease while a genome remains relatively static. Also, according to Morris et al. (2010) "Proteins are more relevant to the biological function of the cell," and proteomic tests can be carried out on serum and urine, which can be obtained quickly and effortlessly, unlike mRNA or DNA (Morris et al., 2010). In clinical proteomics, scientists commonly search for proteins or groups of proteins to help diagnose types of cancers, diseases, or viruses with the goal of early diagnosis. These proteins or groups of proteins can be biomarkers for a disease; see Wulfkuhle et al. (2003), Diamandis (2004), and Visintin et al. (2008). Currently, many of the drugs today work well by targeting proteins, or these drugs are actually proteins themselves. With more advances in proteomics, scientists hope eventually to develop drugs that are made specifically for an individual in order for the drug to be more effective with fewer side effects.

Issues arise in protein identification when an organism's genome has not been sequenced, more specifically in microbial samples. Only 1%-10% of microbes found in the ecosystem can be cultured. There are countless other microbes that have not been identified and, of the microbes that have been cultured, some will show evidence of post translational modifications. These post translational modifications cannot be calculated from the genome (Rose et al., 2010). There has been little progress in the area of envi-

ronmental proteomics. Since only 1%-10% of all microbes can be cultured, being able to correctly identify these microbes via protein identification is of great importance especially in ecological samples such as soil and water samples (Schulze, 2004). Correctly identifying proteins will also aid in the advance of clinical proteomics.

Current methods for identification of proteins are lacking. With a limited number of known genome sequences, noisy data, and incomplete ion sequences, the accuracy of protein identification is limited. In this thesis, we describe a Bayesian approach which aims to improve the identification of proteins.

1.1 Proposed Work

We employ a Bayesian stochastic search approach to protein identification. We use the prior knowledge of abundances of bond cleavages and the probability of any particular amino acid sequence. The likelihood function combines two measures of distance that measure the closeness of each observed m/z value to an m/z value in a theoretical scan of a peptide. An MCMC scheme is utilized to simulate candidate peptides from the posterior distribution, and the peptide with the largest posterior probability is estimated as the true protein. Our approach also allows one to rank the top candidate peptides by their estimated posterior probabilities.

The data comes from the Pacific Northwest National Laboratory (PNNL) and can be publicly accessed online for download (Ansong et al., 2011). This dataset is produced by a LTQ Orbitrap yielding doubly charged tryptic peptides. For each peptide, there is a set of m/z values with corresponding intensity values. Figure 1.1 pictorially shows the spectrum for a given spectrum by plotting the m/z values versus their corresponding intensity values.

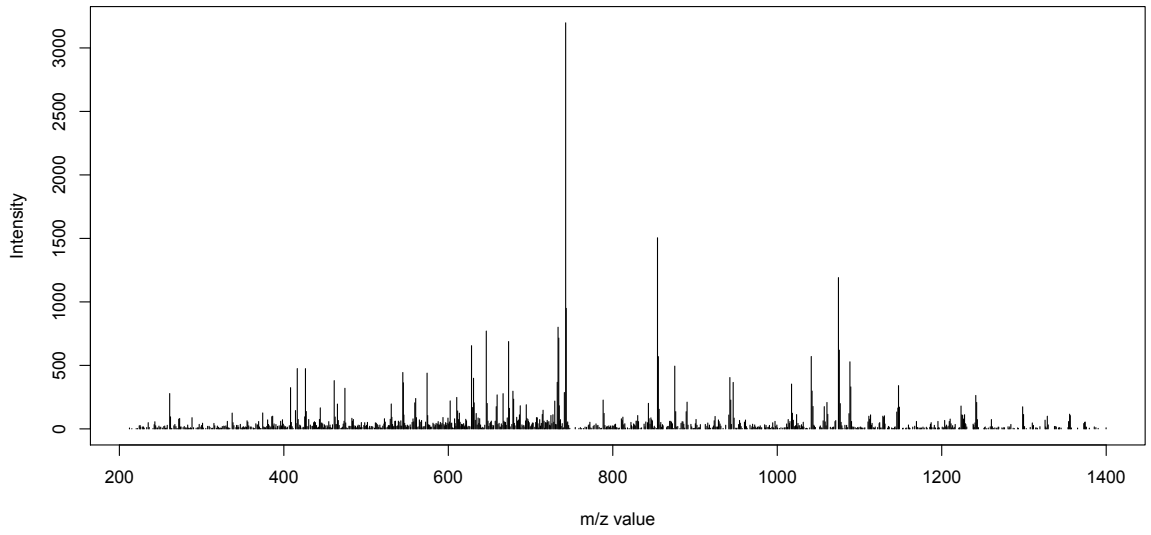

Figure 1.1 Line plot of pairs of intensities and m/z values for a given peptide.

1.2 Overview of Thesis

The thesis is arranged as follows. Chapter 2 describes how one obtains the proteomic profile of a sample and provides insight on current methods for identifying proteins. Chapter 3 introduces basic concepts of peptide fragmentation and the construction of a peptide's theoretical spectrum. Chapter 4 describes our proposed method. Section 4.2 explains our likelihood function. Section 4.3 illustrates the prior knowledge we incorporate into our model and Section 4.4 defines the posterior density. Chapter 5 discusses in detail the Markov chain Monte Carlo algorithm. Chapter 6 illustrates the precision of our method via simulation and Chapter 7 provides a demonstration of our method with a real data application. Chapter 8 concludes the thesis.

CHAPTER 2

MASS SPECTROMETRY AND PROTEIN IDENTIFICATION METHODS

2.1 MASS SPECTROMETRY

There are several methods for obtaining the proteomic profile of a sample. Developed in the 1970s, 2-d gel electrophoresis (2DE) is one of the oldest technologies to identify the proteomic profile. With technological advances, mass spectrometry methods are now more commonly used. In 2DE, proteins from two different samples are analyzed and then the patterns in the proteins are compared. Protein spots that are of interest are then removed from the gel and absorbed by proteolytically or chemically producing peptides that are to be analyzed via mass spectrometry (Issaq et al., 2003).

Matrix-assisted laser desorption and ionization - time of flight mass spectrometry (MALDI-TOF) and surfaced-enhanced laser desorption and ionization mass spectrometry (SELDI-TOF) are two types of mass spectrometry methods. In MALDI-TOF, the proteomic sample must first be mixed with an energy absorbing matrix (EAM). The mixture is then crystallized onto a metal plate. SELDI-TOF includes further chemistry on the target surface of the metal plate, which keeps proteins from complex mixtures according to the distinct properties of the proteins. The metal plate is then placed into a vacuum chamber where the crystallized mixture is hit with pulses from a nitrogen laser. The matrix crystallized molecules consume energy that is produced from the laser and transfer it to the proteins. This causes the proteins to be desorbed and ionized, which creates ions in the gas phase. This process occurs in the presence of an electric field. The electric field speeds up the ions

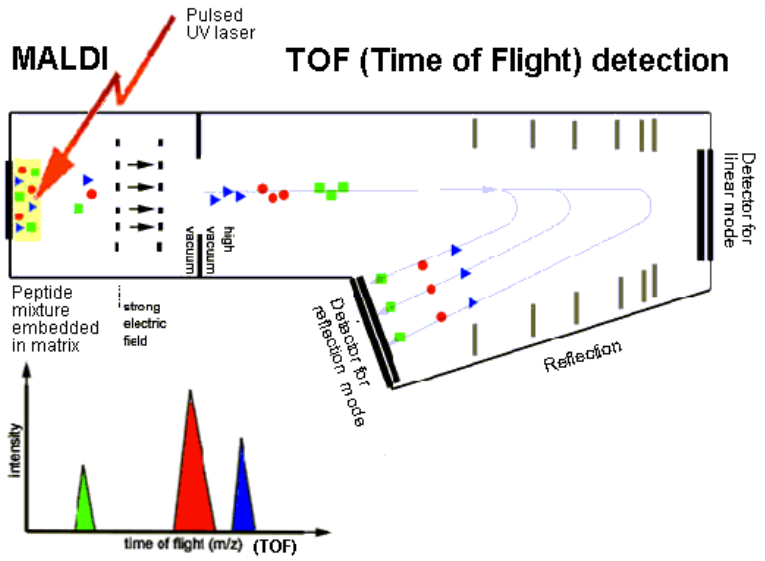

Figure 2.1 A simplified representation of the MALDI-TOF method reproduced by Radboud University Nijmegen (2013).

into a flight tube where the ion hits a detector, which records the time it takes the ion to fly through the tube and strike the detector (Coombes et al., 2007; Issaq et al., 2003). Figure 2.1 provides a simplified illustration of the MALDI-TOF method.

Peaks can be identified by plotting the intensities versus a horizontal index. These peaks characterize the peptide in the current sample. The horizontal index predominantly used in proteomic analysis is the particle's mass-to-charge ratio (m/z). Thus, one can plot the intensities versus the mass-to-charge ratio. In mass spectrometry methods, the m/z value from the time of flight is computed by using a quadratic transformation. To establish the coefficients for the quadratic transformation, a small number of molecules (usually between 3 and 7) with known masses are used to generate a spectrum. Then a count of peaks analogous to the known masses in the spectrum is obtained. Given the set of (time, mass) pairs, the method of least squares determines the coefficients. These calculations are carried out in a preprocessing stage, and the final data spectrum is the line plot of pairs of intensities and m/z values (Coombes et al., 2007).

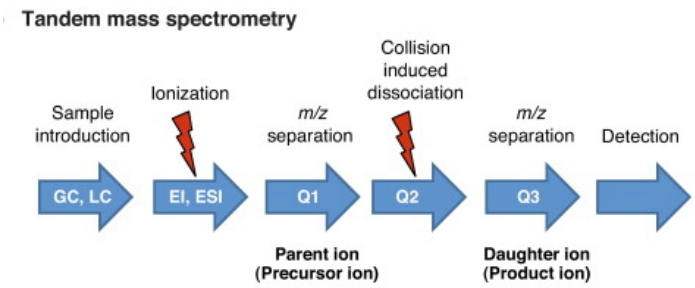

Figure 2.2 This figure, taken from Antoniewicz (2013), shows a simplified representation of how a triple quadrupole mass spectrometer works. Abbreviations: gas chromatography (GC); liquid chromatography (LC); electron impact ionization (EI); electrospray ionization (ESI); first, second, and third quadrupole (Q1, Q2, and Q3).

2.2 Tandem Mass Spectrometry

Tandem mass spectrometry (MS/MS) is a two-stage mass spectrometry process that allows examination of individual ion fragmentation from a group of ions. Figure 2.2 provides a simplified diagram of how a triple quadrupole mass spectrometry works. Tandem mass spectrometry is used with an assortment of instruments and scan modes. There are two key types of instruments used. The first type is an instrument in which two mass spectrometers are assembled in tandem that uses a sequence of mass spectrometers in space, where "in space" means that there is a physical separation of the instrument components. The second type of instrument contains analyzers that store ions where one spectrometer with ion capability assigns a sequence of events in time. The most commonly used tandem mass spectrometer is the triple quadrupole. In a triple quadrupole mass spectrometer, the first and second quadrupole are mass analyzers while the second quadrupole allows ions of any mass to pass through. Figure 2.3 shows a graphical diagram of a triple quadrupole mass spectrometer. A quadrupole mass analyzer is composed of four cylindrical rods that are set parallel to each other. When used in mass spectrometry, the quadrupole is the integral part of the instrument that filters the sample ions based on their m/z values.

There are three scan types that are commonly used: the product ion, precursor ion, and neutral loss scans, where the product ion scan is used most often. In a product ion scan,

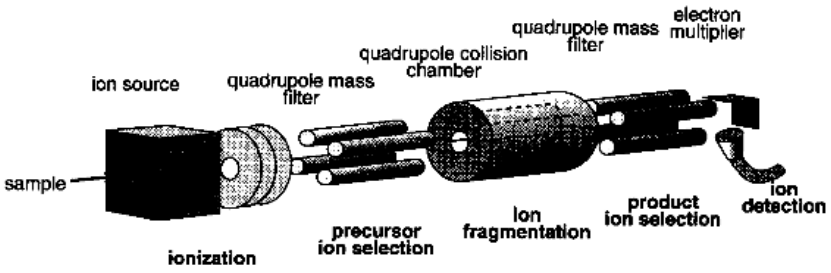

Figure 2.3 This figure, taken from de Hoffmann (1996), shows a schematic diagram of a triple quadrupole mass spectrometer.

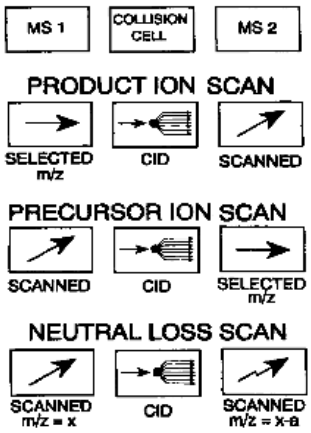

Figure 2.4 This figure, taken from de Hoffmann (1996), graphically shows the difference between the three scan types. MS1 refers to the first mass spectrometer and MS2 refers to the second mass spectrometer.

ions of a specific m/z value are selected in the first mass spectrometer and are analyzed in the second mass spectrometer resulting in a product ion spectrum. In a precursor ion scan, the second mass spectrometer only allows ions with a given m/z value to pass through. The first mass spectrometer, which contains a collision gas, is then scanned over a preset mass range detecting only ions that produce the pre-selected product ion. In a neutral loss scan, the first mass spectrometer scans all the masses while the second mass spectrometer still scans but at a certain offset value from the first mass spectrometer, where the offset value corresponds to a neutral loss that is typically observed for that class of compounds (de Hoffmann, 1996). Figure 2.4 illustrates pictorially the distinction between the three scan modes.

High performance liquid chromatography (HPLC) joined with mass spectrometry is a more recent method for proteomic profiling. HPCL is a separation approach comprising the mass-transfer between the liquid mobile phase and the stationary phase, which could be a solid or a liquid (Standardbase-Techniques, 2012). Although these methods differ in the way they obtain the proteomic profile, they are all extremely sensitive tools.

2.3 Protein Identification Methods

Presently, there are few methods for identifying protein sequences. A popular approach searches through a database of peptides and then matches the closest peptide using the observed spectrum. Some common algorithms for database searches are SEQUEST and MASCOT. Another common approach is de novo sequencing, in which the peptide sequence is determined by recreating a spectrum using the observed spectrum. PepNovo and Peaks are frequently used de novo algorithms. A more recent approach uses a mixture of the other two. In this approach, the de novo method recaptures short peptide sequences and then the peptide sequences are used to refine the search in the database approach (Frank and Pevzner, 2005).

PepNovo is the more commonly used de novo sequencing approach. A scoring function allows one to examine the spectral matches of the candidate peptide and observed spectrum. PepNovo offers two types of scoring methods that typically use a tolerance of 0.5 Da when identifying the ion fragmentation.

The most commonly used score is the ranking score, which is a machine learning ranking algorithm. A discriminative boosting-based method for scoring models was developed by Frank (2009). Their model depends upon a large set of distinct feature functions that compute different qualities of peptide-spectrum matches (PSMs). The method optimizes the models of the PSMs results by differentiating between the correct and incorrect results of the PSMs. Here the boosting-based RankBoost algorithm developed by Freund et al. (2003) is used in the discriminative ranking approach to scoring PSMs. The RankBoost

algorithm is used to train the ranking-based model, which employs a machine learning method called boosting. Several inputs must be supplied to the RankBoost algorithm. The first input is an instance space. When using the rank score method in PepNovo, the instance space is a training set of PSMs from the PepNovo score, which is explained in the next paragraph. The second input is an ordering of the instances from the training set. The last input is a set of feature functions such as the spectrum graph feature and peak annotation features. The purpose of the RankBoost learning algorithm is to compute a ranking function that assigns correct PSMs a higher score than incorrect ones. See Freund et al. (2003) for more details about the RankBoost algorithm and see Frank (2009) for more details about the feature functions. The discriminative boosting-based method is applied to the PepNovo score results and yields the rank score. The peptide with the highest rank is the best estimate for the true peptide (Frank, 2009).

The PepNovo score is another scoring method, which breaks down a probabilistic network in order to develop a better model for peptide fragmentation. Here the objective of the de novo sequencing is to find the peptide that maximizes $P(S|P)$ across all viable candidate peptides where S is the observed spectrum and P is the peptide. There are two continuous values, intensities and cleavage positions, which must be discretized in order to use this scoring method. Due to the fact that the sizes of intensities can be extremely different, k discrete relative intensity levels are assigned to the peaks. A normalized intensity is found by dividing each peak's intensity by the baseline intensity. This baseline is found by averaging the intensities of the weakest 33% of the peaks in the spectrum. Frank and Pevzner (2005) found that $k = 4$ yields the best results. Now the relative cleavage position, $pos(m)$, is discretized into 5 equally sized regions. Here $pos(m)$ is defined as $pos(m) = m/PM$ where m is the position of the cleavage site and PM is the mass of the peptide. A likelihood ratio test is utilized for each prefix mass m of a peptide to show how likely there is to be a cleavage of a peptide at mass m. The peptide prefix masses are the possible cleavage sites, which are the vertices in the spectrum graph. The likelihood ratio

test is composed of two hypothesis tests. The first hypothesis is the collision-induced dissociation (CID) hypothesis. This hypothesis assumes that m is a true cleavage in the peptide that produced the observed spectrum. A probabilistic network is employed to determine the probability $P_{CID}(\vec{I}, m)$ of identifying an experimental series of fragment intensities $\vec{I}$ given that the prefix mass is a cleavage site in the peptide that produced the observed spectrum. The second hypothesis is the random peaks hypothesis. This hypothesis assumes the peaks in the observed spectrum are created by a random process and thus the intensities $\vec{I}$ that correspond to the fragmented ions are assumed to be random. Since each peak is dispersed independently, the probability $P_{Random}(\vec{I}, m)$ is the product of the probabilities of observing the individual peaks in their bins. See Frank and Pevzner (2005) for more details on the CID hypothesis, the random peak hypothesis and spectrum graphs. The logarithm of the likelihood ratio test for a given prefix mass can be defined as

$$Score(\vec{I}, m) = \log \frac{P_{CID}(\vec{I}, m)}{P_{Random}(\vec{I}, m)},$$

where a positive score indicates that the peak intensities, $\vec{I}$, were more likely caused by a true cleavage in the peptide and a negative score indicates that the peak intensities, $\vec{I}$, are more likely to due to random peaks. To obtain the overall score for a peptide, one needs to sum the score for the prefix masses by the following:

$$Score(P) = \sum_{i=1}^{n} Score(\vec{I}_{m_i}, m_i),$$

where P is the candidate peptide. The peptide with the largest score is chosen as the best estimate of the true peptide (Frank and Pevzner, 2005).

Unlike PepNovo and other less common approaches, PEAKS does not convert the spectrum into a graph, but works directly with the spectrum. PEAKS first preprocesses the raw data by reducing the noise and centering the peaks. Since PEAKS works directly with the spectrum, this step is quite important. A score for each mass value is computed for a y and b ion according to the peaks near them. If there are missing peaks for a particular y and b ion pair, a penalty score is given. Next, the set of the 10000 best sequences of

all combinations of amino acids that maximizes the total scores is found. Then the set of 10000 sequences are evaluated further using a more precise scoring function. Finally, a confidence score for each top scoring peptide is produced, and a confidence level for each amino acid residue in the set of top scoring peptides is computed. In a newer version, these sequences are then used to choose promising proteins from a database of proteins based on sequence similarity. Using the same scoring function as the de novo sequencing, each spectrum is compared with all the peptides chosen from the potential proteins (Ma et al., 2003).

In de novo sequencing algorithms, it is important to be able to classify ions correctly and then select informative ions from the observed spectrum to recreate the spectrum in order to identify the peptide. Cleveland and Rose (2012) use a leveraged Bayesian neural network to better classify ions, which leads to better identification of peptides. The details of their work are discussed in more detail in Chapter 8.1.

MASCOT is the most regularly used database protein identification approach. The scoring function that is employed in MASCOT is a molecular weight search (MOWSE), which assesses the match between the observed spectrum and a known peptide. It is assumed that the matches between the peaks of the observed spectrum and the fragmented ions from known peptides are random, and the probability that the matches occur randomly is computed. If the observed spectrum is aligned with the correct peptide, this probability will be extremely small because the peaks in the observed spectrum will match up with the peptide. The scoring function MOWSE only yields scores that indicate the significance of a match. The peptide with the highest score is then chosen as the estimated peptide. In order to get a better understanding of how correct the match is, the score is compared with other peptides.

SEQUEST is another database protein identification approach, which uses a cross-correlation scoring function. Like MASCOT, the scoring function assesses the match between the observed spectrum and a known peptide from the database. Using a simple

model, the theoretical spectrum of peptide is calculated. A displacement value is added to the m/z value of each peak in the spectrum. The correlation is then computed between the observed spectrum and the theoretical spectrum with the displacement value included. The peptide with the highest score is chosen as the estimated peptide (Xu and Ma, 2006).

A major concern of the database search and hybrid method is that they rely on the use of a database of peptides. These methods cannot correctly identify the protein if it is not in the database. Some limitations of de novo peptide sequencing are lack of accuracy and certainty of the chosen peptides, chemical noise, overly complex fragments, and incomplete b and y ion sequences (Lubec and Afjehi-Sadat, 2007). We introduce a Bayesian model that will aim to identify the correct peptide without depending on the database of peptides, but instead using more generic prior information.

CHAPTER 3

BASIC CONCEPTS OF FRAGMENTATION

The basic idea of any protein identification method is to match an observed spectrum to a theoretical spectrum of the proposed peptide. It is extremely difficult to identify intact proteins and so the proteins are broken into short peptides and examined separately. A peptide is a sequence of amino acids, each of which is represented by one of 20 letters. Table 3.1 is a list of all 20 amino acids with their corresponding 3 letter and 1 letter codes. The single letter code is used throughout the thesis to ease notation. The theoretical spectrum of a peptide is a set of peaks with the location of each peak at the m/z value of each ion type. There are spikes at each peak location and zeros everywhere else. The peptide is broken into pairs of ions, most commonly b and y ions. It is the intensities of these ions that are detected in the mass spectrometer. Figure 3.1 shows the theoretical spectrum for the peptide $QVMELLQ$ using just the b and y ions.

Table 3.1 The 20 amino acids with their corresponding 3 letter and 1 letter codes.

Amino Acid	3 Letter Code	1 Letter Code	Amino Acid	3 Letter Code	1 Letter Code
Alanine	Ala	A	Leucine	Leu	L
Arginine	Arg	R	Lysine	Lys	K
Asparagine	Asn	N	Methionine	Met	M
Aspartic Acid	Asp	D	Phenylalanine	Phe	F
Cysteine	Cys	C	Proline	Pro	P
Glutamine	Gln	Q	Serine	Ser	S
Glutamic Acid	Glu	E	Threonine	Thr	T
Glycine	Gly	G	Tryptophan	Trp	W
Histidine	His	H	Tryosine	Try	Y
Isoleucine	Ile	I	Valine	Val	V

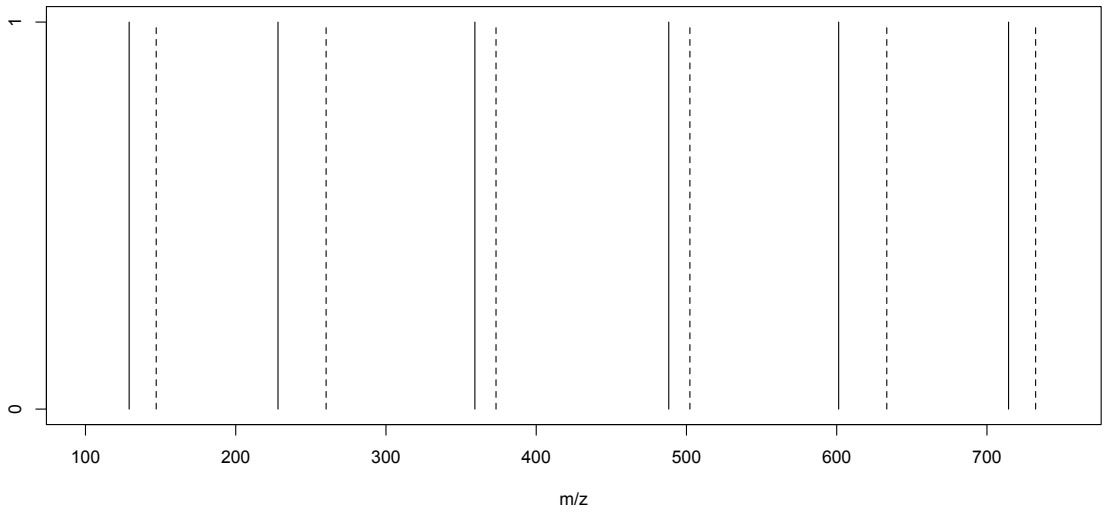

Figure 3.1 Theoretical spectrum for the peptide $QVMELLQ$ using only b and y ions. Here 1 represents the presence of an ion and 0 represents the absence of an ion. The b ion is denoted by solid lines and the y ion is denoted by dashed lines.

To find the theoretical spectrum, one must first split the true peptide sequence into all possible ion combinations. In practice, we use only the b and y ions, although there are several other less common ions. A b ion is the start of the peptide that is terminated by an amino acid with a free amine group $(-NH_2)$. An amine group is any group of organic compounds of nitrogen. Therefore, we classify an ion as a b ion if the charge is maintained on the N-terminus, where the N-terminus refers to the beginning of a peptide that is terminated by an amino acid with a free amine group. In order for an ion to be detected, the ion must have a charge of at least one. The y ion is the complement of the b ion. Thus, it is the end of the peptide where the charge is maintained on the C-terminus, where the C-terminus refers to the end of a peptide that is terminated by a free carboxyl group $(-COOH)$. A carboxyl group is an organic group containing a carbonyl bound to a hydroxyl group where a carbonyl group is a group composed of a carbon atom double bonded to an oxygen atom and a hydroxyl group is group composed of a hydrogen atom covalently bonded to an oxygen atom (**IUBMB**, 1992, p. 48).

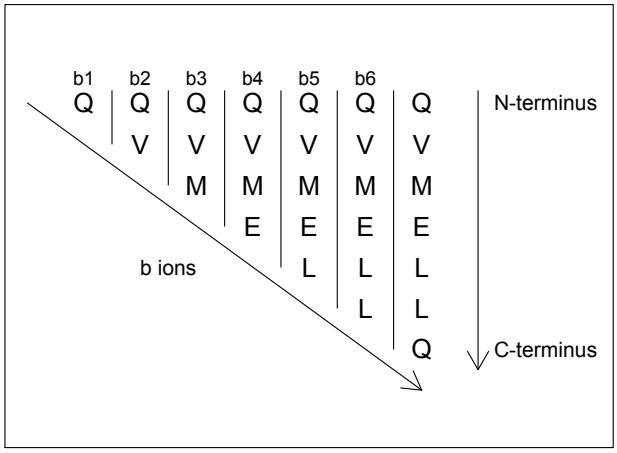

Figure 3.2 Illustration of the fragmentation b ions. Each b ion is on the N-terminus. That is, it is the beginning of the peptide. The first b ion is Q and the last b ion is $QVMELL$. The splitting of the peptide is into all possible binary partitions.

After the b and y ions are found, the mass of each ion is determined. The mass for any given ion is found by $\sum_{i=1}^{k} m(p_i) + \delta_\ell$ where k is the number of amino acids in the ion sequence, p_i is the amino acid in the i th position, $m(p_i)$ is the mass of the amino acid in the i th position, ℓ denotes the type of ion such that $\ell \in \{b, y\}$, and δ_ℓ is the offset for ion type ℓ. Table 3.3 shows a list of all twenty amino acids along with their corresponding mass in Daltons (Da). In tandem mass spectrometry, the peptide fragmentation is determined by offsets that correspond to ion types. That is, the offsets match up to the peaks in a given spectrum, and thus denote the different ion types created in the given mass spectrometer (Dančík et al., 1999). Dančík et al. (1999) develop an offset frequency function to define ion type tendencies for particular mass spectrometers. A common problem was that different types of mass spectrometers yield different spectra. Thus, Dančík et al. (1999) developed an offset frequency function that does not depend on instrument type and allows one to define the ion types produced by a given mass spectrometer. Table 3.2 lists different

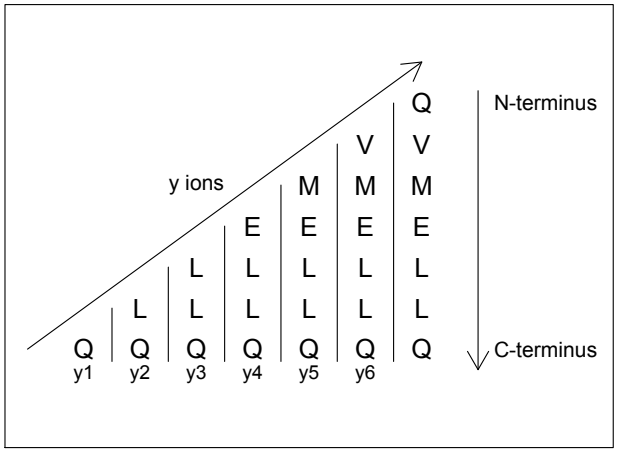

Figure 3.3 Illustration of the fragmentation y ions. Each y ion is on the C-terminus. That is, it is the right-hand end of the peptide. The first y ion is Q and the last y ion is $VMELLQ$. The splitting of the peptide is into all possible binary partitions.

ion types along with the terminus it can be detected from, the offset value, and how to calculate the mass of the ion.

As an example, consider the peptide $QVMELLQ$. There are six b ions and six y ions. The first b ion, Q, has a mass of $128.059 + 0.85 = 128.909$ Da, and the first y ion, Q, has a mass of $128.059 + 18.85 = 146.909$ Da. Continuing with the splitting of the peptide, one obtains the following additional b ions: QV, QVM, $QVME$, $QVMEL$, and $QVMELL$ with masses 227.977, 359.017, 488.060, 601.144, and 714.228 Daltons, respectively. The fragmentation of each b ion can be found in Figure 3.2. Similarly, we obtain the following additional y ions: LQ, LLQ, $ELLQ$, $MELLQ$, and $VMELLQ$ with masses 259.993, 373.077, 502.120, 633.160, and 732.228 Daltons respectively. The fragmentation of each y ion can be found in Figure 3.3. Therefore, the theoretical spectrum for the peptide $QVMELLQ$ is the set of masses: 128.909, 227.977, 359.017, 488.060, 601.144, 714.2284, 732.228, 633.160, 502.120, 373.077, 259.993, and 146.909 Daltons.

Table 3.2 Information about ion types. Here M denotes $\sum_{i=1}^{k} m(p_i)$.

Ion	Terminus	Offset Value	Position
b	N	0.85	(M + 0.85)
b-H_2O	N	-17.05	(M - 17.05)
a	N	-27.15	(M - 27.15)
b-NH_3	N	-16.15	(M - 16.15)
b-H_2O-H_2O	N	-35.20	(M - 35.20)
b-H_2O-NH_3	N	-34.20	(M - 34.20)
a-NH_3	N	-44.25	(M - 44.25)
a-H_2O	N	-45.15	(M - 45.15)
y	C	18.85	(M + 18.85)
y-H_2O	C	0.90	(M + 0.90)
y^2	C	20.05	(M + 20.05)/2
y-NH_3	C	1.90	(M + 1.90)
y^2-H_2O	C	2.30	(M + 2.30)/2
y-H_2O-NH_3	C	-16.10	(M - 16.10)
y-H_2O-H_2O	C	-17.15	(M - 17.15)

Table 3.3 The 20 amino acids with their corresponding masses in Daltons.

Amino Acid	Mass	Amino Acid	Mass
A	71.0371	M	131.04
C	103.009	N	114.043
D	115.027	P	97.0528
E	129.043	Q	128.059
F	147.068	R	156.101
G	57.0215	S	87.032
H	137.059	T	101.048
I	113.084	V	99.0684
K	128.095	W	186.079
L	113.084	Y	163.063

These positions are shown on the (m/z) axis in Figure 3.1.

It is important to find the total mass of the peptide because a mass spectrometer will also measure the total mass of the peptide being analyzed. We can use this weight restriction to eliminate peptides that do not have a total mass within a tolerance of the measured mass. The total mass of the peptide is found by $\sum_{i=1}^{K} m(p_i) +$ mass of H_2O, where the mass of the water is molecule 18.010565 Da. Here K is the number of amino acids in the peptide sequence. For data that are doubly charged, the total mass becomes $\sum_{i=1}^{K} m(p_i) +$ mass of $H_2O + H$ because of the second proton that is acquired. The mass of one hydrogen molecule is 1.00794 Da. Thus the total mass for the peptide $QVMELLQ$ is 860.456 Da assuming the data are doubly charged.

CHAPTER 4

A BAYESIAN MODEL

We propose a Bayesian model with the goal of identifying the true peptide based on the observed spectrum. To identify this true peptide, a Markov chain Monte Carlo (MCMC) algorithm is used to simulate candidate peptide sequences from the posterior distribution. The motivation will be discussed in Section 4.2.

4.1 PRE-PROCESSING

Mass spectrometry data are quite noisy and thus the observed spectrum first needs to be thresholded. In other words, peaks with intensity values below a threshold level will be ignored, and our attention will be focused on the m/z values having intensities above the threshold. A distinct threshold value for each integer m/z value, denoted by $\boldsymbol{T} = (T_1, T_2, \ldots, T_{q\star})$, is computed. Here $q^\star$ denotes the total number of m/z values. Both a constant threshold and a moving threshold are calculated. Then a weighted average of the thresholds is used. For the constant threshold, the c th percentile of the observed intensity values is computed and this computed c th percentile becomes a component of $\boldsymbol{t}$, which is a constant vector of the c th percentile with length $q^\star$. The value of c will typically be chosen to be fairly high (such as 75) so that the only highest intensities are retained.

Consider a peptide whose intensity values range from 1 to 2500 Da with a total of 258 pairs of intensities and m/z values. Using the 75 th percentile, the constant threshold value turns out to be 35.03 and therefore, each element of $\boldsymbol{t}$ would be 35.03. If one uses just a constant threshold, then the data used in our method would be a set of m/z and intensity values where the intensity values would be greater than or equal to 35.03.

The mass spectrometer does not always capture all peaks at the beginning and the end of the spectrum. Thus using only a constant threshold could remove peaks that are truly signal peaks and not noise peaks. Therefore, we threshold using a combination of constant and moving thresholds.

A moving threshold is then calculated as follows:

1. Consider any fixed m/z value x^*.
2. A subsection of the m/z values is selected using a window width of 50 Daltons on either side of x^*.
3. The c th percentile of the observed intensity values within the window around x^* is found.
4. This is done for a fine grid of m/z values, and each computed c th percentile becomes a component of $\boldsymbol{t}' = (t'_1, t'_2, \ldots, t'_{q^\star})$.

Now consider the same peptide whose intensity values range from 1 to 2500 Da with a total of 258 pairs of intensities and m/z values. Here each element of $\boldsymbol{t}'$ will be the 75 th percentile of each observed instensity within a window of x^*. For this example, the elements of $\boldsymbol{t}'$ range from 14.25 to 124.81.

Now, using both the constant and moving thresholds, a weighted average is then found.

1. A sequence of weights, denoted as $\boldsymbol{v}$, is computed having initial component 0.999 and with equally-spaced components decreasing linearly until the middle component of 0.25 and then increasing until the last component of 0.999. This serves as the weight vector for the constant threshold.
2. $(1 - \boldsymbol{v})$ is then the sequence of weights for the moving threshold.
3. $\boldsymbol{v} \circ \boldsymbol{t}$ and $(1 - \boldsymbol{v}) \circ \boldsymbol{t}'$ are computed and denoted ν_1 and ν_2, respectively, where $\boldsymbol{v} \circ \boldsymbol{t}$ gives the elementwise product here.

4. The elementwise sum of ν_1 and ν_2 is found and each value becomes a component of $\boldsymbol{T}$, i.e. $\boldsymbol{T} = \nu_1 + \nu_2$.

5. Each observed m/z value is matched up with the nearest corresponding threshold value within $\boldsymbol{T}$.

6. If the observed m/z value has a corresponding observed intensity value that is above the threshold value in $\boldsymbol{T}$, then the observed intensity value is retained. Otherwise, the observed intensity value and corresponding observed m/z value is removed.

The data used in our method contain the retained intensity values and their corresponding m/z values. Figures 4.1, 4.2, and 4.3 illustrate how thresholding reduces the noisy peaks in the spectrum for different peptides. Figure (a) in each plot shows the full observed spectrum before thresholding and Figure (b) in each plot shows the observed spectrum after thresholding. Figure (a) in the Figures 4.1, 4.4, and 4.5 show the full observed spectrum before thresholding for the same peptide while Figure (b) shows the observed spectrum after thesholding using a different value for the c th percentile. One can see that the smaller the value of c, the more peaks remain in the spectrum. As the value of c becomes larger, the fewer peaks remain in the spectrum. Thus, the value of c is important and must be chosen correctly so that it does not allow too many noise peaks or does not eliminate too many signal peaks from the observed spectrum.

4.2 LIKELIHOOD

For our Bayesian model, we first specify a likelihood function, which gives a measure of how well the observed spectrum and theoretical spectrum agree. If a candidate peptide's theoretical spectrum does not align well with the observed spectrum, an overall goodness of fit measure is computed that will penalize the candidate peptide. However, if the theoretical spectrum does align nicely with the observed spectrum, the overall goodness of fit measure will reward the candidate peptide. Even after thresholding, we still expect there to be

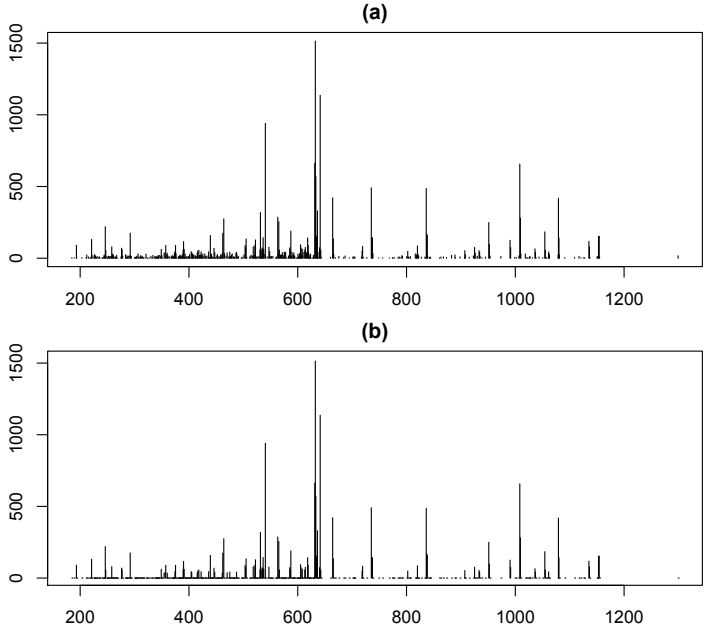

Figure 4.1 Figure (a) shows an observed spectrum before thresholding. Figure (b) shows the observed spectrum after thresholding. The c th percentile was set to be 75% when calculating the threshold.

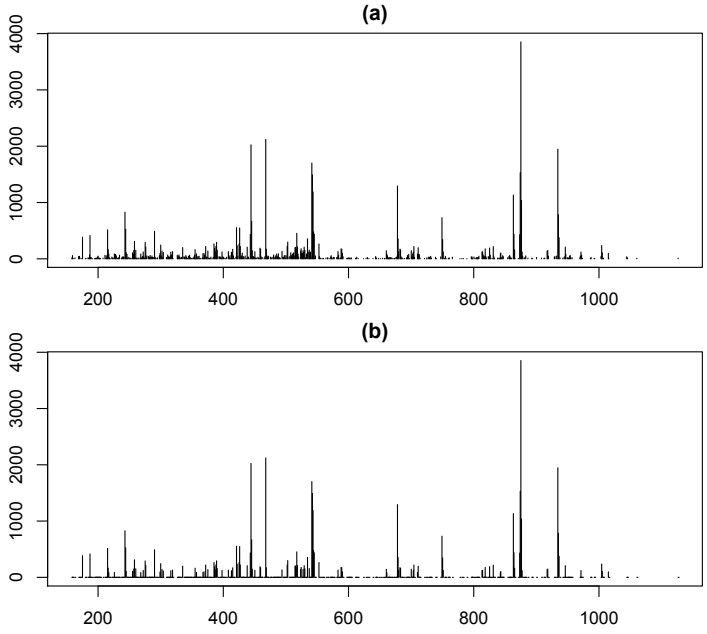

Figure 4.2 Figure (a) shows an observed spectrum before thresholding. Figure (b) shows the observed spectrum after thresholding. The c th percentile was set to be 75% when calculating the threshold.

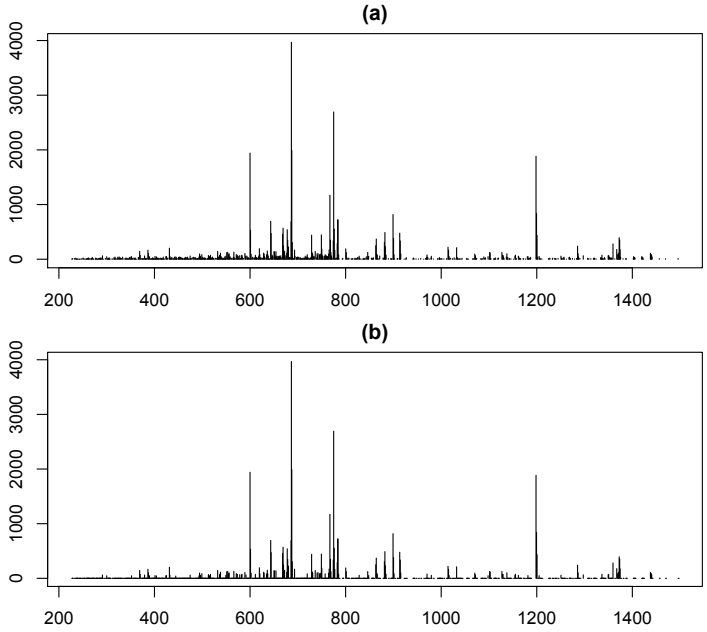

Figure 4.3 Figure (a) shows an observed spectrum before thresholding. Figure (b) shows the observed spectrum after thresholding. The c th percentile was set to be 75% when calculating the threshold.

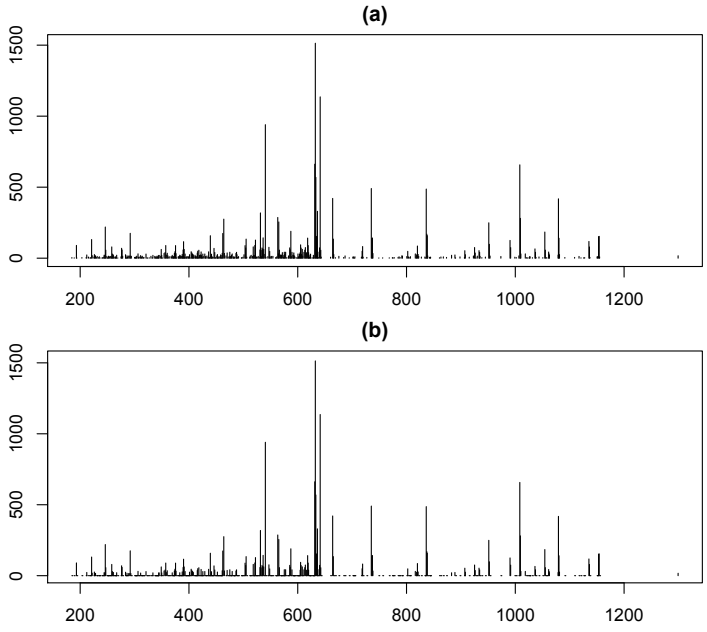

Figure 4.4 Figure (a) shows an observed spectrum before thresholding. Figure (b) shows the observed spectrum after thresholding. The c th percentile was set to be 50% when calculating the threshold.

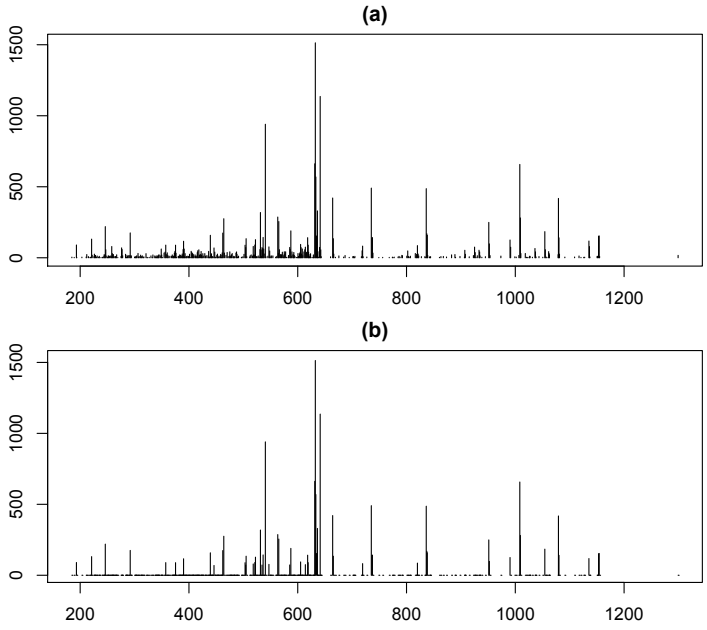

Figure 4.5 Figure (a) shows an observed spectrum before thresholding. Figure (b) shows the observed spectrum after thresholding. The c th percentile was set to be 90% when calculating the threshold.

noise peaks in the data set and therefore, we incorporate another overall goodness of fit measure that will penalize a candidate peptide when there are too many noise peaks near the theoretical spectrum and reward a candidate peptide when there are not. We do know that the mass spectrometer does not always capture every signal peak. Hence, we include an indicator function in our likelihood function that signifies the presence or absence of a peak.

We propose a likelihood function of the form

$$L(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{\eta}, \kappa_1, \kappa_2) \propto \kappa_1^s \exp(-\kappa_1 S_1) \kappa_2^{t-s} \exp(-\kappa_2 S_2) \tag{4.1}$$

where our parameter vector $\boldsymbol{\theta} = (\tau_1^b, \ldots, \tau_p^b, \lambda_1^b, \ldots, \lambda_p^b, \tau_1^y, \ldots, \tau_p^y, \lambda_1^y, \ldots, \lambda_p^y)$ and $\boldsymbol{X}$ contains the observed pairs of m/z values and intensities for a particular spectrum and $\boldsymbol{\eta}$ represents the string of amino acids for the candidate peptide. The other parameters are explained below. The size of the likelihood function is driven by two overall goodness of fit measures S_1 and S_2, defined as:

$$S_1 = \sum_{i=1}^{p} \left[\lambda_i^b \min_j |x_j - \tau_i^b| + \lambda_i^y \min_j |x_j - \tau_i^y| \right] \tag{4.2}$$

and

$$S_2 = \sum_{j=1}^{t} \min_{i,k} |x_j - \tau_i^k| \tag{4.3}$$

where $k \in \{b, y\}$, x_j are the observed m/z values that have corresponding observed intensity values above $\boldsymbol{T}$, and τ_i^b and τ_i^y are the m/z values for the b and y ion of the candidate peptide, respectively, for $i = 1, \ldots, p$. Here, p represents the number of b ions (or, equivalently, the number of y ions) and t represents the number of m/z values that have an intensity above the threshold $\boldsymbol{T}$. Also, κ_1 and κ_2 represent weights where s is the number of b and y ions combined. Here λ_i^b and $\lambda_i^y \in \{0, 1\}$ are indicator functions that signify whether the i th b or y ion has a corresponding observed peak, where $i = 1, \ldots, p$. Here, λ_i^b = 1 denotes the presence of a b ion at position i and λ_i^b = 0 denotes the absence of a b ion. Similarly, λ_i^y = 1 denotes the presence of a y ion at position i and λ_i^y = 0 denotes the absence of a y ion.

Here, S_1 measures the sum of minimum absolute distances between the closest observed m/z above a threshold and each m/z peak value of the candidate peak value, while S_2 measures the sum of minimum absolute distances between each observed m/z value above a threshold and the closest candidate peak m/z values. That is, S_1 measures the closeness of the nearest observed peak to each candidate b ion or y ion, and S_2 measures the closeness of the nearest candidate peak to each observed peak. When all peaks for the candidate peptide are very close to observed peaks that are above the threshold, then $\exp(-S_1)$ is high. When all the observed peaks are close to candidate peaks, $\exp(-S_2)$ will be high. One can think of $\exp(-S_1)$ and $\exp(-S_2)$ in terms of sensitivity and specificity. Table 4.1 illustrates this analogy. Therefore, $\exp(-S_1)$ is high when the false negative rate is low and $\exp(-S_2)$ is high when the false positive rate is low.

Table 4.1 This table shows relationship between $\exp(-S_1)$ and $\exp(-S_2)$ and sensitivity and specificity. Thus, when the true positive rate is high ($\exp(-S_1)$ is high), this corresponds to a high sensitivity rate. When the true negative rate is high ($\exp(-S_2)$ is high), this corresponds to a high specificity rate.

		Candidate True Peptide Sequence	
		Peak	No Peak
Observed Spectrum	Peak	True Positive	False Positive
	No Peak	False Negative	True Negative

4.3 PRIORS

Huang et al. (2004) estimated the average bond cleavage abundance for each amino acid pair for both the b and y ions for gas-phase dissociation spectra. Many studies involving protein identification obtain gas-phase dissociation spectra. Collision-induced dissociation (CID) fragments the peptides even further: When an ion collides "with a gaseous target, energy is redistributed among different vibrational degrees of freedom within the ion" (Wysocki et al., 2006, p. 283). Further fragmentation leads to formation of charged ions (Wysocki et al., 2006). There is limited knowledge about unimolecular dissociation (Huang et al., 2004). Huang et al. (2004) shed light on some of the factors that affect unimolecular dissociation by exploring cleavage abundance for both b and y ions. A cleavage occurs when the peptide bond fragments during collision induced dissociation. Therefore, a cleavage pair is the b and y ion pair that are present in the peptide. For example, take the peptide $QVMELLQ$. Recall from Chapter 3 that QV is one of the six b ions of the peptide $QVMELLQ$ and the complement to that b ion is the y ion $MELLQ$. These complementary ions are a result of the cleavage between the amino acids V and M. This information from Huang et al. (2004) will give us insight about when we expect to see cleavages in the pairs of amino acid residues, and thus we use this information to develop prior information about cleavage pair abundance for our Bayesian approach to identify the true peptide.

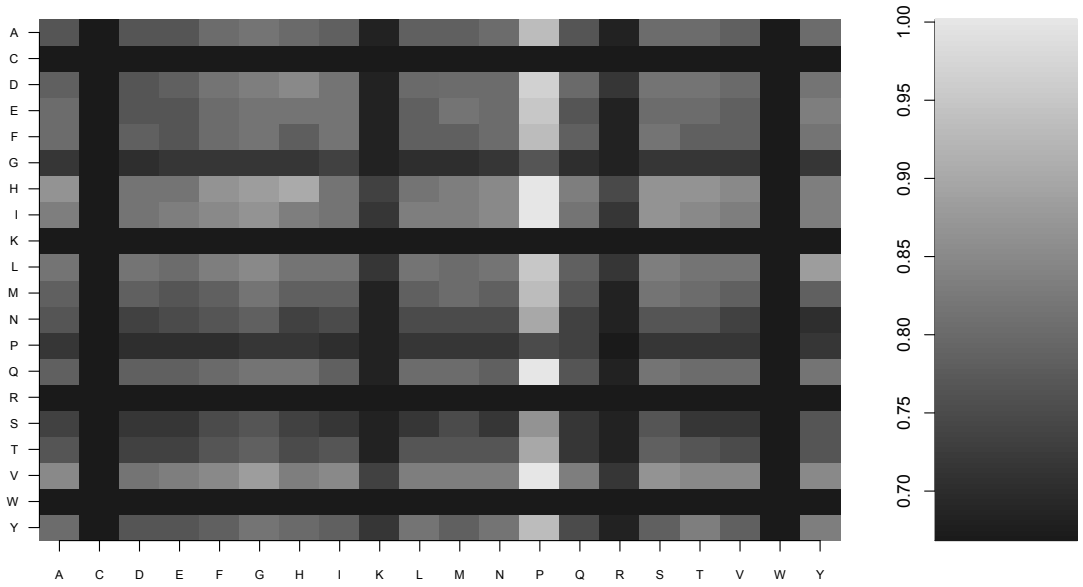

Figure 4.6 A display of the geometric mean of average relative abundances of bond cleavages of b and y ions for any particular amino acid pair using Figure 1 in Huang et al. (2004). Note that this is a linear transformation of the scale used in Huang et al. (2004). The linear transformation is of the form $\rho = 0.49x + 0.67$. The y-axis is the the single letter code of the amino acid on the N-terminal amino acid and the x-axis is the single letter code of the amino acid on the C-terminal amino acid.

Cleavage Prior

The cleavage pair abundance prior is denoted $\pi(\boldsymbol{\lambda}|\boldsymbol{\beta}, \boldsymbol{\gamma})$. For notational simplicity we write $\pi(\boldsymbol{\lambda})$ since $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are fixed, and then $\pi(\boldsymbol{\lambda})$ is defined as:

$$\pi(\boldsymbol{\lambda}) = \prod_{i=1}^{p} P(\lambda_i^b, \lambda_i^y) \tag{4.4}$$

with

$$\begin{aligned}
P(\lambda_i^b = \lambda_i^y = 1) &= \rho_i^{by} \times \gamma_i \times \beta_i \\
P(\lambda_i^b = 1, \lambda_i^y = 0) &= \rho_i^{by} \times (1 - \gamma_i) \times \beta_i \\
P(\lambda_i^b = 0, \lambda_i^y = 1) &= \rho_i^{by} \times \gamma_i \times (1 - \beta_i) \\
P(\lambda_i^b = \lambda_i^y = 0) &= 1 - \rho_i^{by} + [\rho_i^{by} \times (1 - \gamma_i) \times (1 - \beta_i)]
\end{aligned}$$

where $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^b, \boldsymbol{\lambda}^y) = (\lambda_1^b, \ldots, \lambda_p^b, \lambda_1^y, \ldots, \lambda_p^y)$, ρ_i^{by} is the geometric mean of the average relative abundance of bond cleavages of b and y ions for a particular amino acid pair for $i = 1, \ldots, p$ derived from Huang et al. (2004), γ_i is the probability of the presence of a y ion, and β_i is the probability of the presence of a b ion. Note that we use a linear transformation of the probability $\rho = 0.49x + 0.67$, as explained below. Here, p represents the number of cleavage pairs. Note that the λ_i^b's are modeled as having random marginal Bernoulli distributions with probabilities $\rho_i^{by}\beta_i$ and the λ_i^ys are modeled as having random marginal Bernoulli distributions with probabilities $\rho_i^{by}\gamma_i$, and λ_i^b, λ_i^y are all mutually independent for $i = 1, \ldots, p$. The proof of the λ_i^b's and λ_i^y's having a marginal Bernoulli distribution is shown below. Let $P(\lambda_i^b = x_b, \lambda_i^y = x_y)$.

$$\begin{aligned} P(\lambda_i^b = 1) &= \sum_x P(\lambda_i^b = 1, \lambda_i^y = x) \\ &= \rho_i^{by}\gamma_i\beta_i + \rho_i^{by}(1-\gamma_i)\beta_i \\ &= \rho_i^{by}\beta_i \end{aligned}$$

$$\begin{aligned} P(\lambda_i^b = 0) &= \sum_x P(\lambda_i^b = 0, \lambda_i^y = x) \\ &= \rho_i^{by}\gamma_i(1-\beta_i) + 1 - \rho_i^{by} + \rho_i^{by}(1-\gamma_i)(1-\beta_i) \\ &= \rho_i^{by}(1-\beta_i) + 1 - \rho_i^{by} \\ &= 1 - \rho_i^{by}\beta_i \end{aligned}$$

The derivation for the λ_i^y's can shown below:

$$\begin{aligned} P(\lambda_i^y = 1) &= \sum_x P(\lambda_i^b = x, \lambda_i^y = 1) \\ &= \rho_i^{by}\gamma_i\beta_i + \rho_i^{by}(1-\beta_i)\gamma_i \\ &= \rho_i^{by}\gamma_i \end{aligned}$$

$$\begin{aligned} P(\lambda_i^y = 0) &= \sum_x P(\lambda_i^b = x, \lambda_i^y = 0) \\ &= \rho_i^{by}\beta_i(1-\gamma_i) + 1 - \rho_i^{by} + \rho_i^{by}(1-\gamma_i)(1-\beta_i) \\ &= \rho_i^{by}(1-\gamma_i) + 1 - \rho_i^{by} \\ &= 1 - \rho_i^{by}\gamma_i. \end{aligned}$$

As an example, consider the peptide $QVMELLQ$. From Chapter 3, we know that there are six b and y ions. Therefore, we will require $\boldsymbol{\lambda}^b$ and $\boldsymbol{\lambda}^y$ each to have six elements. Suppose we set the probabilities for the presence of each ion to be contained in vectors $\boldsymbol{\beta} = [0.05, 0.9, 0.9, 0.9, 0.9, 0.75]$ and $\boldsymbol{\gamma} = [0.1, 0.9, 0.9, 0.9, 0.9, 0.75]$. The probabilities can be different for each element, as illustrated. These types of values for the $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ vectors are motivated in Chapter 7. For example, suppose we have $\boldsymbol{\lambda}^b = [0, 1, 1, 1, 1, 0]$ and $\boldsymbol{\lambda}^y = [0, 1, 1, 1, 0, 1]$. The first cleavage pair for the peptide $QVMELLQ$ is QV. Continuing with the splitting of the peptide, one obtains the following cleavage pairs: VM, ME, EL, LL, and LQ. Using Figure 4.6, we obtain the vector of geometric mean of the average relative abundance of bond cleavages of b and y, $\boldsymbol{\rho}^{by} = (\rho_1^{by}, \ldots, \rho_6^{by}) = (0.802, 0.835, 0.769, 0.785, 0.818, 0.785)$. Figure 4.6 shows the geometric mean of the average bond cleavage abundance for all cleavage pairs of the b and y ions using Figure 1 in Huang et al. (2004). Note that probabilities for a particular amino acid cleavage pair that are too close to zero may force the algorithm to exclude reasonable peptides. In order for our prior to be more inclusive, we use a linear transformation of the scale used in Huang et al. (2004). The linear transformation is of the form $\rho = 0.49x + 0.67$. Our rescaled

distribution has probabilities that range from 0.67 to 1.00. The cleavage pair abundance prior probability for the example above is given by

$$P(\lambda_1^b = \lambda_1^y = 0) = 1 - 0.802 + [0.802 \times (1 - 0.1) \times (1 - 0.05)] = 0.884$$
$$P(\lambda_2^b = 1, \lambda_2^y = 1) = 0.835 \times 0.9 \times 0.9 = 0.676$$
$$P(\lambda_3^b = 1, \lambda_3^y = 1) = 0.769 \times 0.9 \times 0.9 = 0.623$$
$$P(\lambda_4^b = 1, \lambda_4^y = 1) = 0.785 \times 0.9 \times 0.9 = 0.636$$
$$P(\lambda_5^b = 1, \lambda_5^y = 0) = 0.818 \times (1 - 0.9) \times 0.9 = 0.074$$
$$P(\lambda_6^b = 0, \lambda_6^y = 1) = 0.785 \times 0.75 \times (1 - 0.75) = 0.147$$
$$\pi(\lambda) = \prod_{i=1}^{6} P(\lambda_i^b, \lambda_i^y) = 0.0029, \text{ and}$$
$$\log_e \pi(\lambda) = -5.84.$$

As a matter of notation, note that our parameter vector $\boldsymbol{\theta}$ $(= \boldsymbol{\theta}_{\gamma,\,\beta})$ depends on the values of γ and β, but our notation will suppress this dependency since γ and β will remain fixed throughout the algorithm.

Sequence Prior

We now want to specify a prior distribution for a particular sequence (or string) of amino acids in a peptide. The probability of any particular amino acid sequence is represented by the string prior, $\pi(\boldsymbol{\eta})$, which quantifies the probability of a sequence of amino acids appearing consecutively in a peptide sequence. For each amino acid pair in the candidate peptide under consideration, we count how often the pair occurs in the set of known peptides from the same species. Then we find the empirical probability of each amino acid pair using our large database of peptides. Note that one could use other databases that do not contain the current peptide to calculate the empirical probability. The string prior is defined as

$$\pi(\boldsymbol{\eta}) = \sqrt{\pi(\boldsymbol{\eta}_F) \times \pi(\boldsymbol{\eta}_R)}, \tag{4.5}$$

where $\pi(\boldsymbol{\eta}_F)$ is the joint probability of any particular amino acid sequence calculated from left to right while $\pi(\boldsymbol{\eta}_R)$ is the joint probability of any particular amino acid sequence calculated in the reverse direction. Note that this is the geometric mean of $\pi(\boldsymbol{\eta}_F)$ and $\pi(\boldsymbol{\eta}_R)$. Here $\boldsymbol{\eta}$ is the ordered sequence of the amino acids in the current peptide under consideration, and $\pi(\boldsymbol{\eta})$ is a probability for this particular sequence. Denote a generic peptide sequence by $A_1 A_2 \cdots A_{m-1}$, where $m-1$ is the number of amino acids in the peptide sequence, A_0 denotes the beginning of the sequence, and A_m denotes the end of the sequence. For example, consider the peptide $TGMSNVSK$. For this candidate peptide having 8 amino acids, $m = 9$. $\pi(\boldsymbol{\eta}_F)$ is calculated by

$$\pi(\boldsymbol{\eta}_F) = P(A_1 = a_1) \times \prod_{i=1}^{m-1} P(A_{i+1} = a_{i+1} | A_i = a_i) \tag{4.6}$$

with $P(A_1 = a_1) = p_1$, $P[(A_i, A_{i+1}) = (a_i, a_{i+1})] = p_{i,i+1}$, and therefore $P(A_{i+1} = a_{i+1} | A_i = a_i) = \frac{p_{i,i+1}}{\sum_j P[(A_i, A_{i+1} = (a_i, j)]}$ for $j \in \{A, C, \ldots, Y, \_\}$ where a_i represents the amino acid in the i th position in the peptide sequence and a_m = _ signifies the termination of a sequence. In a similar manner, $\pi(\eta_R)$ is computed by

$$\pi(\boldsymbol{\eta}_R) = P(A_{m-1} = a_{m-1}) \times \prod_{i=0}^{m-2} P(A_i = a_i | A_{i+1} = a_{i+1}) \tag{4.7}$$

with $P(A_{m-1} = a_{m-1}) = p_{m-1}$, $P(A_i = a_i | A_{i+1} = a_{i+1}) = \frac{p_{i,i+1}}{\sum_j P[(A_i, A_{i+1} = (j, a_{i+1})]}$ for $j \in \{A, C, \ldots, Y, \_\}$ where a_0 = _ signifies the beginning of a peptide sequence.

We illustrate the calculation of this type of prior probability using the peptide $TGMSNVSK$ from the salmonella typhimurium species. First note that the empirical probability that the initial amino acid in a peptide is T is 0.0614. The empirical probability that a random ordered pair of amino acids is TG is 0.0067, and so on. Figure 4.7 shows the joint empirical probabilities for all pairs of amino acids. The probability of the particular amino acid

sequence for the peptide $TGMSNVSK$ in the forward direction, $\pi(\boldsymbol{\eta}_F)$, is found by:

$$\begin{aligned}
P(A_1 = T) &= 0.0614 \\
P(A_2 = G|A_1 = T) &= \frac{P[(A_1, A_2) = (T, G)]}{P[(A_1, A_2) = (T, A)] + \cdots + P[(A_1, A_2) = (T, _)]} \\
&= \frac{p_{1,2}}{P[(A_1, A_2) = (T, A)] + \cdots + P[(A_1, A_2) = (T, _)]} \\
&= \frac{0.0067}{0.0500 + \cdots + 0.0019} = 0.0992 \\
&\vdots \\
P(A_8 = K|A_7 = S) &= \frac{0.0019}{0.0654} = 0.0295 \\
P(A_9 = _\,|A_8 = K) &= \frac{0.5146}{0.5239} = 0.9823,
\end{aligned}$$

yielding

$$\pi(\boldsymbol{\eta}_F) = P(A_1 = T) \times \textstyle\prod_{i=1}^{9} P(A_{i+1} = a_{i+1}|A_i = a_i) = 4.26 \times 10^{-11} \text{ and}$$
$$\log_e \pi(\boldsymbol{\eta}_F) = -23.88.$$

The probability for the reverse direction, $\pi(\boldsymbol{\eta}_R)$, is calculated similarly and is shown below:

$$\begin{aligned}
P(A_8 = K) &= 0.5146 \\
P(A_7 = S|A_8 = K) &= \frac{P[(A_7, A_8) = (S, K)]}{P[(A_7, A_8) = (A, K)] + \cdots + P[(A_7, A_8) = (_, K)]} \\
&= \frac{p_{1,2}}{P[(A_7, A_8) = (A, K)] + \cdots + P[(A_7, A_8) = (_, K)]} \\
&= \frac{0.0019}{0.0059 + \cdots + 0.0107} = 0.0130 \\
&\vdots \\
P(A_1 = T|A_2 = G) &= \frac{0.0067}{0.1229} = 0.0545 \\
P(A_0 = _\,|A_8 = T) &= \frac{0.0614}{0.1226} = 0.5008,
\end{aligned}$$

yielding

$$\pi(\boldsymbol{\eta}_R) = P(A_1 = T) \times \textstyle\prod_{i=0}^{8} P(A_i = a_i|A_{i+1} = a_{i+1}) = 1.31 \times 10^{-12} \text{ and}$$
$$\log_e \pi(\boldsymbol{\eta}_R) = -27.36.$$

Therefore, we have $\pi(\boldsymbol{\eta}) = \sqrt{4.26 \times 10^{-11} \times 1.31 \times 10^{-12}} = 7.47 \times 10^{-12}$ and so $\log_e \pi(\boldsymbol{\eta}) = -25.62$.

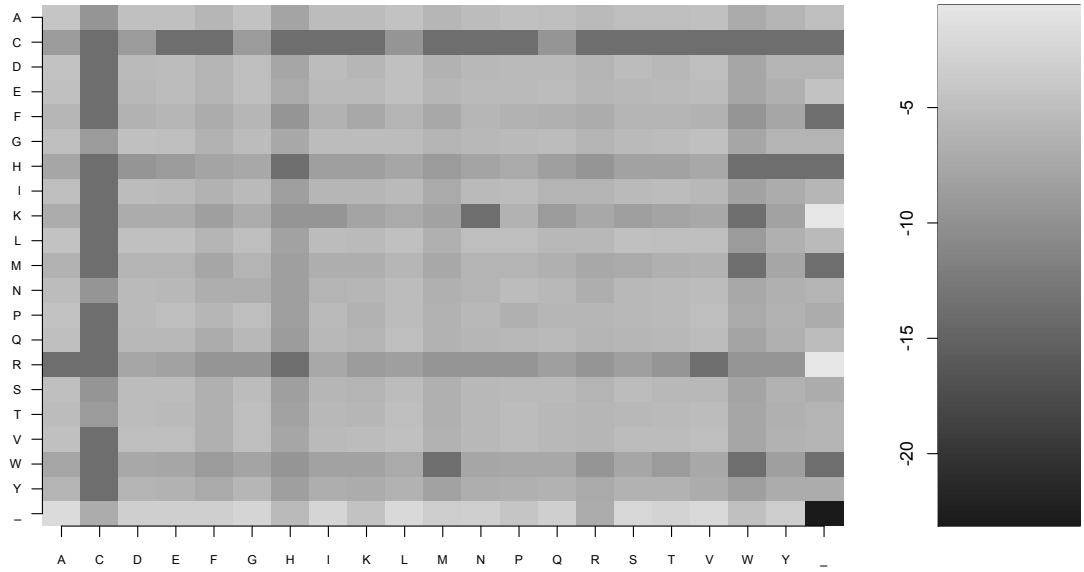

Figure 4.7 The joint empirical probabilities for all pairs of amino acids. For ease of identifying the different joint empirical probabilities, the figure is shown on the log-scale. The y-axis is the single letter code of the amino acid for the first amino acid in the pair and the x-axis is the single letter code of the amino acid for the second amino acid in the pair. The darker the square, the less probable the pair.

Prior for κ_1, κ_2

The concentration parameters, κ_1 and κ_2, are assumed to have independent Gamma(a_1, b_1) and Gamma(a_2, b_2) distributions respectively, which are independent of the other parameters.

4.4 POSTERIOR

Using Bayes' Theorem, the posterior density can be written as

$$\pi(\boldsymbol{\eta}, \boldsymbol{\lambda}, \kappa_1, \kappa_2 | \boldsymbol{X}) \propto L(\boldsymbol{X} | \boldsymbol{\lambda}, \boldsymbol{\tau}, \boldsymbol{\eta}, \kappa_1, \kappa_2) \times \pi(\boldsymbol{\lambda}) \times \pi(\boldsymbol{\eta}, \boldsymbol{\tau}) \times \pi(\kappa_1, \kappa_2) \tag{4.8}$$

$$= L(\boldsymbol{X} | \boldsymbol{\theta}, \boldsymbol{\eta}, \kappa_1, \kappa_2) \times \pi(\boldsymbol{\lambda}) \times \pi(\boldsymbol{\eta}) \times \pi(\kappa_1, \kappa_2), \tag{4.9}$$

where $\boldsymbol{\lambda}$, $\boldsymbol{\eta}$, and κ_1, κ_2 are assumed independent. Note that the set of m/z locations given by $\boldsymbol{\tau} = (\tau_1^b, \ldots, \tau_p^b, \tau_1^y, \ldots, \tau_p^y)^T$ are determined by the sequence $\boldsymbol{\eta}$, and so $P(\boldsymbol{\tau} | \boldsymbol{\eta}) = 1$. Note that this posterior density is only known up to a constant and the actual form of the posterior density is complicated. Therefore, to obtain the posterior probabilities we must use MCMC simulation. Our Bayesian method incorporates prior information about the chance of seeing particular cleavage pairs, and also quantifies the prior probability of any particular specific amino acid sequence. We use this posterior density to estimate the true peptide, with candidate peptides having high posteriors being judged more likely to be the true peptide. Our point estimate of the true peptide is the posterior mode, that is, the candidate peptide (among those visited by the search algorithm) with the highest posterior probability, and the posterior distribution variance provides information about the uncertainty of the estimate.

CHAPTER 5

A MARKOV CHAIN MONTE CARLO ALGORITHM

5.1 MARKOV CHAIN MONTE CARLO

There are various circumstances in which we wish to sample from a particular distribution but it can be extremely difficult especially when the normalizing constant is unknown. Markov chain Monte Carlo simulations allow one to sample from these complex distributions. A Markov chain is a sequence of random variables $\{X_n; n \geq 0\}$ with an invariant distribution π that is constructed from a transition kernel. A random process is considered to be a Markov process if the transition probabilities between the different values in the state space depend only on the current state. A transition kernel is a function P defined by $P(X_n, A) = P\{X_{n+1} \in A | X_0, \ldots, X_n\}$ such that for all measurable sets A, $\pi(A) = \int \pi(dx)P(x, A)$. A Markov chain is considered to be irreducible if for an initial state there is a positive probability that it will visit every state in the state space. That is, one can move from any state to any other state. A Markov chain with a finite number of values is considered to be aperiodic if the greatest common divisor of return times to any particular state is 1 (Tierney, 1994; Robert and Casella, 1999; Andrieu et al., 2003; Sorensen and Gianola, 2002).

Markov chain Monte Carlo (MCMC) refers to methods for sampling from an invariant distribution based upon the construction of a Markov chain. The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970; Andrieu et al., 2003; Sorensen and Gianola, 2002) is a type of MCMC method that obtains a sequence of random samples with a complex invariant distribution. The Metropolis-Hastings algorithm begins with a target density

π. Then a conditional density $q(y|x)$ is chosen and the algorithm produces a Markov chain $\{X^{(t)}\}$ through a Markov transition. Here q is called the proposal distribution. The transition is described as follows for a given $x^{(t)}$:

1. Generate $Y_t \sim q(y|x^{(t)})$

2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \min\left\{\dfrac{\pi(Y_t)}{\pi(x^{(t)})}\dfrac{q(x^{(t)}|Y_t)}{q(Y_t|x^{(t)})}, 1\right\} \\ x^{(t)} & \text{with probability} \quad 1 - \min\left\{\dfrac{\pi(Y_t)}{\pi(x^{(t)})}\dfrac{q(x^{(t)}|Y_t)}{q(Y_t|x^{(t)})}, 1\right\} \end{cases}$$

Because the Hastings ratio only depends on the ratio of target densities, the complicated normalizing constant need not be known.

The Gibbs Sampler (Geman and Geman, 1984; Gelfand and Smith, 1990) is a special case of the Metropolis-Hastings where the random value is always accepted and only univariate full conditional distributions are considered. A univariate distribution implies that all the random variables have fixed values but one. One then simulates n random variables sequentially from the n univariate conditionals in order to produce a sample from the full joint distribution (Robert and Casella, 1999).

5.2 INITIALIZATION

We first need to find a starting peptide for the MCMC algorithm. We will only consider candidates with the overall correct mass (within a tolerance) when generating an initial candidate. One option is to use an initial iterative sub-algorithm to obtain a starting peptide. Note the actual mass of the true peptide is available to us from the mass spectrometry data, and so we can dramatically reduce the parameter space by searching for peptides with a mass within a specific tolerance of the actual mass, which we take as $\pm$ 0.5 Da. Here is the algorithm for obtaining the random starting point.

INITIALIZATION ALGORITHM

1. The weight of the true peptide is given.

2. A random sample of either one, two, or three amino acids is selected as the initial guess, $\boldsymbol{\eta}_{curr}$, for the starting peptide and the mass of the resulting peptide is found.

3. Let Δ be the weight of the true peptide and Δ_{curr} be the weight of the current peptide. Then $r_1 = |\Delta - \Delta_{curr}|$ is calculated.

4. If r_1 is less than a tolerance of 0.5 Daltons, then $\boldsymbol{\eta}_{curr}$ becomes the starting peptide. However, if $r_1 \geq 0.5$, then we randomly add or remove either one, two, or three amino acids creating a new peptide, $\boldsymbol{\eta}_{new}$.

5. For each addition or removal, the weight Δ_{new} of the resulting new peptide is found and r_2 is calculated where

$$r_2 = \exp\left(-\left(\,|\Delta - \Delta_{new}| - |\Delta - \Delta_{curr}|\right)\right). \tag{5.1}$$

6. r_2 is then compared to $U \sim U(0,1)$.

7. If $r_2 > U$, then $\boldsymbol{\eta}_{new}$ becomes $\boldsymbol{\eta}_{curr}$, otherwise $\boldsymbol{\eta}_{curr}$ is unchanged.

8. Go to 3.

This algorithm continues until a peptide is found having a weight within 0.5 Da of the true peptide's weight. To illustrate this algorithm, consider the true peptide $AQLQEIQTK$ having a total mass of 1058.59 Da. After running the algorithm with that total mass, the random starting peptide is $GPLHAPWGSH$, having a total mass of 1058.52 Da, which is within 0.5 Da of the total mass of the true peptide.

While using the method above will reduce the space of initial peptides, it may still yield a starting peptide far from the truth if the peptide sequence is long, which could result in our method taking a long time to search for the true peptide.

Another option for finding a starting peptide is to use the results from PepNovo. PepNovo yields a list of the top 2000 best estimated peptides for the true peptide. We can use

a peptide from this list as our starting peptide. We still ensure the starting peptide will have the correct total mass within a tolerance. Here we provide an example of how we use a starting peptide from the PepNovo results. Consider the same true peptide $AQLQEIQTK$. We then choose a random protein from the list of the top 2000 estimated peptides from PepNovo. Some of the estimated peptides may not have the correct total mass within a tolerance, and so we must choose a peptide that satisfies our weight constraint. Therefore, a starting peptide could be, for example, $KALQNQAQTQ$.

5.3 Posterior Simulation

Once the starting peptide is generated, the log likelihood of that peptide is calculated using the likelihood defined above. Let us call the current peptide $\boldsymbol{\eta}_{curr}$ (initially this will be the starting peptide). The $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ vectors are pre-determined at the beginning of the algorithm and are constant throughout the algorithm. Before the algorithm begins, a vector $\boldsymbol{\lambda}_{curr}$ is generated using the $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ vectors.

MARKOV CHAIN MONTE CARLO ALGORITHM

1. A new peptide is created by randomly replacing one, two, or three amino acids of the current peptide with one, two, or three amino acids.

2. Generate κ_1 from its full conditional distribution, a Gamma distribution with the shape parameter $\alpha_1 = a_1 + s$ and scale parameter $\beta_1 = S_1 + b_1$. Similarly, generate κ_2 from its full conditional distribution, a Gamma distribution with the shape parameter $\alpha_2 = a_2 + (t - s)$ and scale parameter $\beta_2 = S_2 + b_2$. Here, a_1, b_1, a_2, and b_2 are some constants chosen as parameters of the priors for κ_1 and κ_2. Note that the values of S_1 and S_2 are based on the current peptide.

3. The log likelihoods of the current peptide and new peptide are then found and denoted as ℓ_{curr} and ℓ_{new}, respectively.

4. Generate $\boldsymbol{\lambda}_{new}$ using $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

5. Calculate $\pi(\boldsymbol{\lambda}_{new})$, $\pi(\boldsymbol{\lambda}_{curr})$, $\pi(\boldsymbol{\eta}_{new})$, and $\pi(\boldsymbol{\eta}_{curr})$ up to a constant.

6. The posterior probability is computed for both the new and current peptide. Denote these as ζ_1 and ζ_2, respectively.

7. The proposal densities $q(\lambda_{curr}|\lambda_{new})$, $q(\lambda_{new}|\lambda_{curr})$, $q(\boldsymbol{\eta}_{curr}|\boldsymbol{\eta}_{new})$, and $q(\boldsymbol{\eta}_{new}|\boldsymbol{\eta}_{curr})$ are calculated.

8. Generate $U \sim U(0,1)$. If $U < \left(\frac{\zeta_1}{\zeta_2} \times \frac{q(\lambda_{curr}|\lambda_{new})}{q(\lambda_{new}|\lambda_{curr})} \times \frac{q(\boldsymbol{\eta}_{curr}|\boldsymbol{\eta}_{new})}{q(\boldsymbol{\eta}_{new}|\boldsymbol{\eta}_{curr})}\right)$, then the new peptide becomes the current peptide, and $\boldsymbol{\lambda}_{new}$ becomes $\boldsymbol{\lambda}_{curr}$. Otherwise, both the current peptide and $\boldsymbol{\lambda}_{curr}$ remain unchanged.

9. Go to 1.

When exploring large state spaces stochastically, it is important that the algorithm be irreducible. That is, we must ensure it may visit every potential state with positive probability (Tierney, 1994). In order to ensure irreducibility, every 1000 steps we generate an entirely new peptide that is independent of the current state. Note that any sequence with the correct mass has positive probability of being generated in this step (Tierney, 1994).

Steps 1 - 9 are repeated for a specific number of iterations (e.g., $25,000$). The peptide with the largest posterior density is selected as the true peptide, and we retain all generated peptides along with their approximate posterior probabilities (up to a constant).

We argue that our state space is finite, because for any given spectrum, the peptide cannot be arbitrarily long. Stochastic search algorithms with finite state spaces typically satisfy certain theoretical properties more readily than those with an infinite number of states (Tierney, 1994). A mass spectrometer will always accurately measure the total weight of the true peptide and thus, there is a maximum number of residues that will produce a peptide of that weight. The amino acid G has the smallest mass of 57.0215. Therefore, a string of all G's would be the longest peptide for a required mass.

Trace plots of the log posterior and parameters will be used to monitor convergence of the algorithm. Trace plots are plots of specific sampled values versus a simulation index.

Here our simulation index is the number of iterations. This plot allows one to see whether the chain has converged to its stationary distribution. Another use of a trace plot is to see whether the chain is mixing well. A chain that mixes well will cross its posterior space quickly (SAS Institute Inc., 2009).

To calculate the first proposal densities we need to calculate $q(\boldsymbol{\lambda}_{curr}|\boldsymbol{\lambda}_{new})$ and $q(\boldsymbol{\lambda}_{new}|\boldsymbol{\lambda}_{curr})$. Note that $q(\boldsymbol{\lambda}_{curr}|\boldsymbol{\lambda}_{new}) = q(\boldsymbol{\lambda}_{curr})$ and $q(\boldsymbol{\lambda}_{new}|\boldsymbol{\lambda}_{curr}) = q(\boldsymbol{\lambda}_{new})$ since the new $\boldsymbol{\lambda}$ is generated independently of the current $\boldsymbol{\lambda}$. Recall that $\boldsymbol{\lambda}$ follows a Bernoulli distribution. Suppose $\boldsymbol{\Lambda}$ is a multinomial random variable and $\boldsymbol{\lambda}$ is a particular vector. Suppose we set $\boldsymbol{\beta} = (0.05, 0.99, 0.99)$ and $\boldsymbol{\gamma} = (0.10, 0.99, 0.99)$ and suppose we have $\boldsymbol{\lambda}_{curr} = (0, 1, 1, 0, 1, 1)$ and $\boldsymbol{\lambda}_{new} = (0, 0, 1, 1, 0, 1)$. We would calculate the proposal densities as follows:

$$\begin{aligned} q(\lambda_{curr}|\lambda_{new}) = q(\lambda_{curr}) &= P(\Lambda_{curr} = \boldsymbol{\lambda}_{curr}) \\ &= (0.95) \times (0.99) \times (0.99) \times (0.90) \times (0.99) \times (0.99) \\ &= 0.82 \end{aligned}$$

and

$$\begin{aligned} q(\lambda_{new}|\lambda_{curr}) = q(\lambda_{new}) &= P(\Lambda_{new} = \boldsymbol{\lambda}_{new}) \\ &= (0.95) \times (0.01) \times (0.99) \times (0.10) \times (0.01) \times (0.99) \\ &= 9.3 \times 10^{-6}. \end{aligned}$$

To calculate the second set of proposal densities we need to calculate $q(\boldsymbol{\eta}_{curr}|\boldsymbol{\eta}_{new})$ and $q(\boldsymbol{\eta}_{new}|\boldsymbol{\eta}_{curr})$. Recall from step 1 of the $MCMC$ algorithm, we are always replacing either one, two, or three amino acids of the current peptide with either one, two, or three amino acids. Hence we know that there is a $1/3$ chance that we will choose either one, two, or three amino acids to be replaced. If only one animo acid is chosen to be replaced, then there is a $1/n$ chance that any particular amino acid will be chosen. Here n represents the

total number of amino acids in the peptide sequence. If a pair of amino acids is chosen to be replaced, then there is a $1/(n-1)$ chance that a consecutive pair of amino acids will be chosen. If three consecutive animo acids are chosen to be replaced, then there is a $1/(n-2)$ chance that any particular triplet of consecutive amino acids will be chosen.

Also, note that the current and new peptide must have a total mass that is within a tolerance of the total mass of the true peptide. After the number of amino acids to be replaced is fixed, a list of single, pairs, and/or triplets of amino acids is selected that each have a mass within a tolerance of the mass of the amino acid(s) that is to be replaced. Therefore, the probability that a particular single, pair, or triplet is chosen is $1/m$ where m is the number of singles, pairs, and/or triplets in the list of amino acids that satisfy the weight tolerance. If a pair or triplet is selected from the list, then we must consider all permutations of the pair or triplet. For example, if the pair AK is selected from the list, we then randomly select whether AK or KA is chosen. The probability for choosing a particular permutation of a pair of amino acids is $1/2$. The probability for choosing a particular permutation of a triplet of amino acids is $1/q$ where q is the number of permutations of the three amino acids. Since there are no permutations for a single amino acid, the probability is 1.

To illustrate the calculation of the proposal probabilities, let us look at an example. Consider the current peptide to be $TGMSNVSK$ and the new peptide to be $TGMSNWK$ with a tolerance level set to be 0.5. In order for the current peptide to be replaced by the new peptide, the pair of amino acids VS needs to be replaced with the single amino acid W. There is a $1/3$ probability that a set of two amino acids is chosen to be replaced. Since two amino acids are chosen to be replaced, the probability we choose the pair of amino acids VS is $1/(8-1) = 1/7$. The mass of VS is 186.1 and thus, we can only replace the pair of amino acids with either one, two, or three amino acids that have a mass within $186.1 \pm$ the tolerance. Given that the tolerance is 0.5, we can only replace VS with AD, EG, W, and SV. Therefore, the probability that W is selected is $1/4$. Thus, $q(\boldsymbol{\eta}_{new}|\boldsymbol{\eta}_{curr}) = (1/3) \times (1/7) \times (1/4) \times 1 = 0.012$. Now, in order for the new peptide

to be replaced by the current peptide, the amino acid W needs to be replaced with the pair of amino acids VS. There is a $1/3$ probability that one amino acid is chosen to be replaced. Since one amino acid is chosen to be replaced, the probability we choose the amino acid W is $1/7$. The mass of W is 186.079 and thus, we can only replace the single amino acid with either one, two, or three amino acids that have a mass within $186.079 \pm$ tolerance. Given that the tolerance is 0.5, we can only replace W with AD, EG, W, and SV. Therefore, the probability that SV is selected is $1/4$. Now we need to look at the permutations of SV, which are SV and VS. The probability that VS is chosen is $1/2$. Thus, $q(\boldsymbol{\eta}_{curr}|\boldsymbol{\eta}_{new}) = (1/3) \times (1/7) \times (1/4) \times (1/2) = 0.006$.

CHAPTER 6

SIMULATION STUDY

In this chapter, we simulate data based on our likelihood in order to get a better understanding of the tuning parameters, with the goal of recovering the theoretical spectrum more often. One approach could be to generate a full spectrum with intensities at each m/z location. Then one could pick out the m/z values that are above our threshold. Since our algorithm uses only the m/z values that have intensities above a threshold, we will instead generate a spectrum with signal and noise peaks that are already assumed to be above a threshold. For a given peptide, we will know the locations of the true peaks. Let us denote the true set of peak locations as $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_s)$, where s represents the total number of true peaks. Each true peak will then generate a signal peak and a random number of noise peaks that are above the threshold. Here we look at two different noise structures, one using the Laplace distribution and the other using a Poisson process.

6.1 LAPLACE NOISE STRUCTURE

We first employ the Laplace distribution to simulate noise peaks (Damsleth and El-Shaarawi, 1989; Kemp, 2003). Using the Laplace distribution ensures that we have a generative model. A generative model randomly generates observable data by fixing a joint probability distribution over observation sequences. The basic idea of a generative model is to model the data directly or aid in composing a conditional probability function, which is formed through Bayes' rule (Singla and Domingos, 2005). A generative model allows us to generate spectra using our model that is defined in Chapter 4.

Mass spectrometers do not always capture peaks that appear at the beginning or end

of the spectrum, causing the rate of noise peaks per signal peak to vary over an observed spectrum. Therefore, before we generate a spectrum, we first split the observed spectrum into three sections. Each section will contain a number of signal peaks, determined to be a percentage of the total signal peaks. Then for each signal peak in each section, a random number of noise peaks will be generated. To explain how to find the number of signal peaks for each section, consider a peptide with $s = 20$ true peaks. The first section will contain $s_1 = 20 \times 0.1 = 2$ signal peaks and the third section will contain $s_3 = 20 \times 0.1 = 2$ signal. Thus the middle section will contain $s_2 = 20 - 2 - 2 = 16$ signal peaks. The proportions of 0.1 for each boundary section were chosen based upon extensive numerical experimentation. For each section of the spectrum, we use a discrete uniform with parameters $a = 0$ and b to determine the number of noise peaks per signal peaks to be generated. The values of b depend upon the section of the spectrum. Lower values of b will be chosen for the beginning and ending sections and a higher value of b will be chosen for the middle section. Note that increasing b for each section will cause our data to become noisier. For the first section, the value of b, denoted as b_1, will be $b_1 = 3$. For the middle and third section the value of b will be $b_2 = 10$ and $b_3 = 5$, respectively. These values work well after many analyses and tend to generate a moderate number of noise peaks.

We then simulate from a Laplace distribution to generate the locations of both the signal peak and noise peaks with fixed parameters κ_1 and κ_2. Figure 6.1 shows the plot of generated signal peaks versus the true signal peaks (theoretical spectrum) for a given peptide using different values of κ_1. This illustrates how the choice of κ_1 affects the location of the generated signal peaks. The choice of κ_1 for Figure 6.1 (a) is $\kappa_1 = 50$, $\kappa_1 = 1000$ for Figure 6.1 (b), and $\kappa_1 = 0.05$ for Figure 6.1 (c). Increasing κ_1 does not have much effect on the location of the generated signal peaks. However, upon decreasing κ_1, the location of the generated signal peaks are shifted from their location on the theoretical spectrum. To illustrate how the choice of κ_2 affects the location of the generated noise peaks, consider the peptide $TGMSNVSK$. Figure 6.2 shows the plot of a generated spectrum versus the

observed spectrum corresponding to the peptide $TGMSNVSK$ and theoretical spectrum for a given peptide using different values of κ_2. This shows how the choice of κ_2 affects the location of the generated noise peaks. The choice of κ_2 for Figure 6.2 (a) is $\kappa_2 = 0.10$, $\kappa_2 = 1.0$ for Figure 6.2 (b), and $\kappa_2 = 0.01$ for Figure 6.2 (c). Increasing κ_2 causes the location of the generated noise peaks to be tightly centered on the signal peaks. Decreasing the value of κ_2 causes the location of the generated noise peaks to spread out far from the signal peaks.

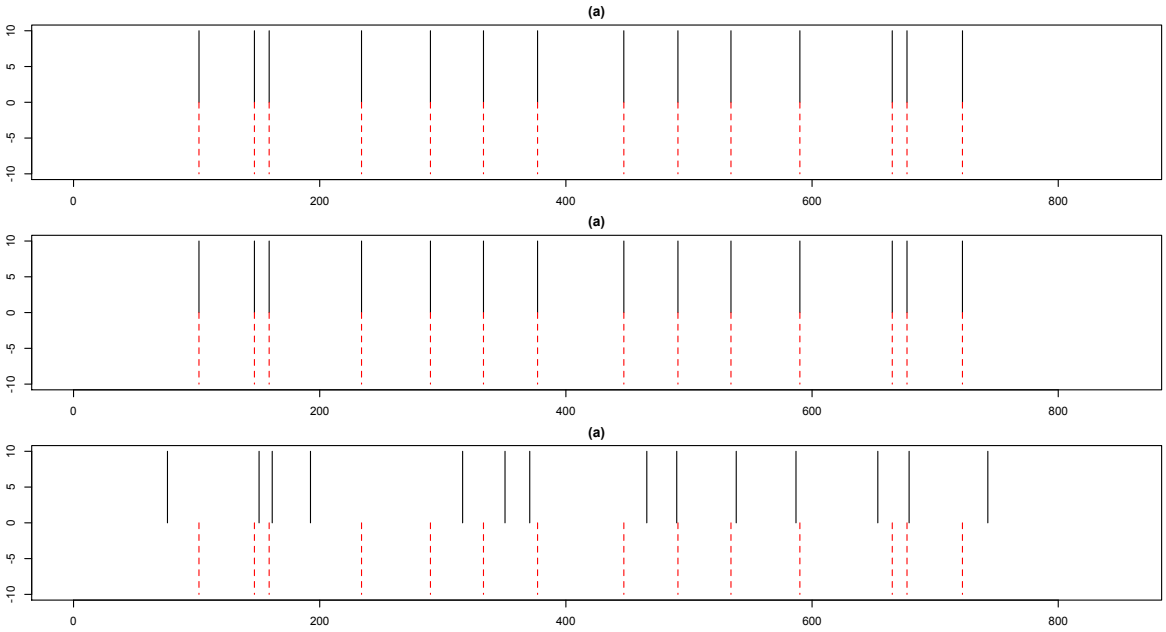

Figure 6.1 Simulated signal peaks plotted against the theoretical spectrum. The simulated signal peaks are plotted above the zero axis. The theoretical spectrum is plotted with the dashed lines below the zero axis. Values for κ_1: (a) $\kappa_1 = 50$, (b) $\kappa_1 = 1000$, (c) $\kappa_1 = 0.05$.

The steps to generate a spectrum with Laplace noise structrue is as follows:

1. Determine the total number of true peaks, s, and compute $\boldsymbol{\tau}$ by finding the b and y ions, based on the given peptide.

2. Simulate signal peaks from a density $f(x_i) \propto e^{-\kappa_1|x_i-\tau_i|}$ for $i = 1, \ldots, s$.

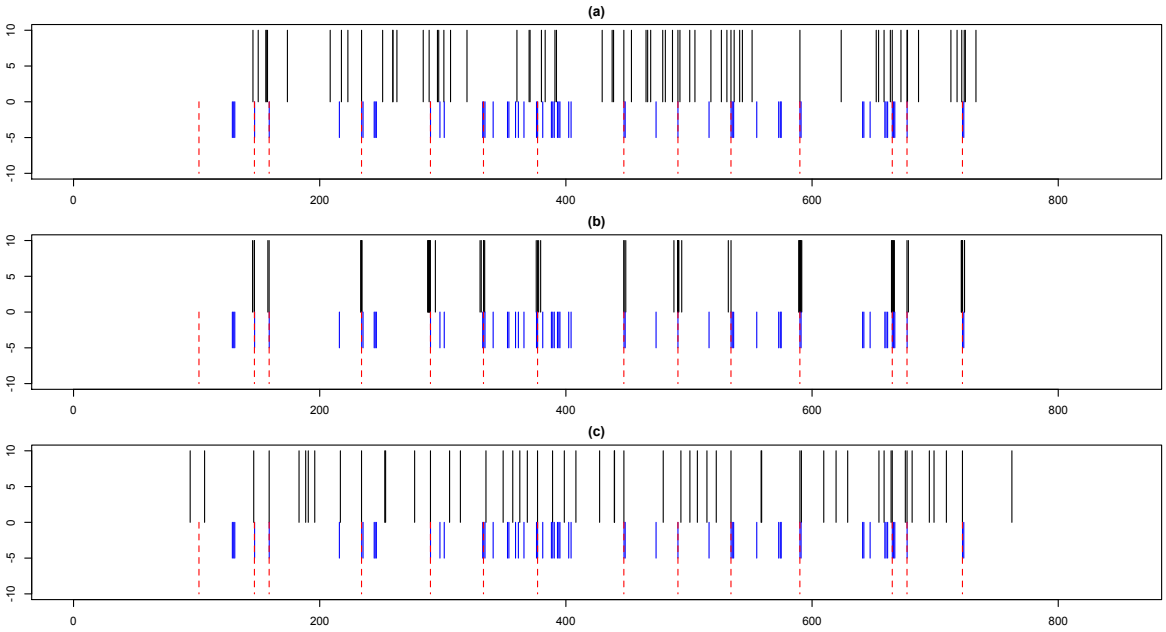

Figure 6.2 Simulated spectrum plotted against the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is plotted with the solid lines below the zero axis and the theoretical spectrum is plotted with the dashed lines below the zero axis. Values for κ_2: (a) $\kappa_2 = 0.10$, (b) $\kappa_2 = 1.0$, (c) $\kappa_1 = 0.01$.

3. Use an indicator function λ_i with probability function $P(\boldsymbol{\lambda}) = \prod_{i=1}^{p} P(\lambda_i^b, \lambda_i^y)$, where $p = s - 1$ is the total number of b ions (or, equivalently, the number of y ions) to determine the presence or absence of each signal peak. $P(\boldsymbol{\lambda})$ was given in Section 4.3.

4. For each of the three sections in the spectrum, the number of noise peaks for each signal peak in the section is generated using a discrete uniform.

5. Simulate noise peaks from a density $f(x_j) \propto e^{-\kappa_2 |x_j - \tau_i|}$, where t represents the total number of peaks for $j = s + 1, \ldots, t$.

The likelihood for the generated set of peaks is of the form

$$L \propto \kappa_1 e^{-\kappa_1 \sum_{i=1}^{s} \lambda_i \ |x_i - \tau_i|} \kappa_2^{t-s} e^{-\kappa_2 \sum_{j=1}^{t} |x_j - \tau_j|}. \tag{6.1}$$

Note the similarity of this expression with the previous likelihood in Equation 4.1 discussed Section 4.2, although it is not exactly the same. The first component in the likelihood defined in Section 4.2 sums over the minimum absolute distances between the closest observed peak to a candidate peak and the second component sums over the minimum absolute distance between the nearest candidate peak to each observed peak. In Equation 6.1, the first component just sums over the absolute distances between the closest observed peak to a candidate peak and the second component sums over the absolute distance between the nearest candidate peak to each observed peak. In the following simulations, we use the previous likelihood in Equation 4.1, not Equation 6.1, for inference.

Example 6.1

Before simulating the spectrum, we must specify the parameters. We set κ_1 and κ_2 to be 50 and 0.10, respectively. For the indicator function λ, we set the first elements of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ to be $p_{b1} = 0.05$ and $p_{y_1} = 0.10$. We set these probabilities to be low because the mass spectrometer rarely captures the first b ion and first y ion and thus we want to ensure that our simulated data represent the observed spectrum well. We set all other p_{bi} and p_{y_i} to equal 0.80 for $i = 2, \ldots, p$.

We first consider a peptide with a short amino acid sequence. Consider the peptide $TGMSNVSK$ whose observed spectrum contains m/z values that range from 123 to 749 Da. The total number of true peaks is $s = 14$ and so $s_1 = 1$, $s_2 = 12$, and $s_3 = 1$ using boundary section proportions 0.1.

For each example shown (Examples 6.1, 6.2, 6.3, 6.4), a table of the top estimated peptides is given along with their corresponding log posterior value. The breakdown of the log posterior is also given to see which part of the model is most heavily influencing the log posterior and to see why the true peptide is not estimated to the best (if that is the case). The top estimated candidates are chosen as a percentage of unique candidate peptides accepted by the algorithm. To illustrate how the top estimated candidates are chosen, consider that if

the algorithm has 100 unique peptides that were accepted by the algorithm, then there will be $100 \times 0.10 = 10$ peptides (with the largest log posteriors) selected as the top estimated peptides.

Figure 6.3 shows the plot of the simulated data versus the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines. One can see the simulated spectrum mimics the observed spectrum reasonably well. After the spectrum is generated, our method described in

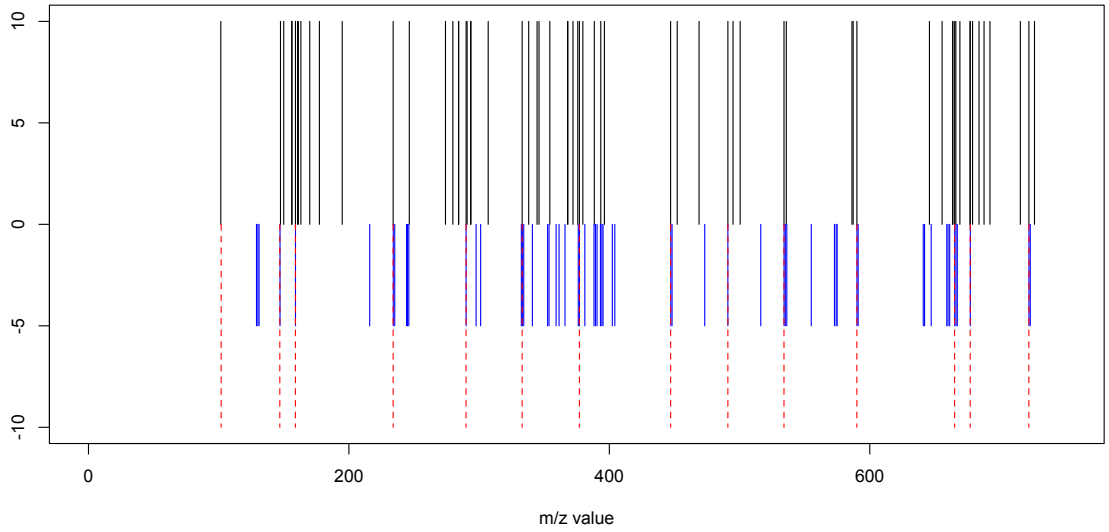

Figure 6.3 Simulated spectrum plotted against the observed spectrum and theoretical spectrum when using the Laplace noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Chapter 4 is then applied to the simulated spectrum. The starting peptide is $GTMSGRSQ$, which was obtained from the results from PepNovo when applied to the real data. The algorithm was run for 10000 iterations and the best estimated peptide for the true peptide is $TGMSGGVSK$ with a log posterior density of 28.18 (up to a constant). Table 6.1 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum

Table 6.1 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
TGMSGGVSK	28.18	58.57	-7.11	-28.31	5.41	-0.37
TGMSGGEGK	28.06	59.21	-9.14	-27.11	5.41	-0.30
TGMSNVSK	18.95	45.25	-5.60	-25.62	5.34	-0.41
TSTSNVSK	15.43	43.82	-7.97	-25.39	5.47	-0.50
SAMSNVSK	13.87	41.49	-6.85	-25.66	5.38	-0.50
TGFGKVSK	11.74	43.12	-8.82	-27.65	5.46	-0.37
SDSSNVSK	7.61	34.67	-7.33	-23.84	4.47	-0.37

along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Here it is the third best peptide.

To ensure our algorithm is performing accurately, we simulate a spectrum with minimal noise. To decrease the number of noise peaks, we set the values of b for each section of the spectrum to be $b_1 = b_2 = b_3 = 2$. We also generate a spectrum with substantial noise by increasing the values of b for each section to $b_1 = b_2 = b_3 = 10$. Our method was then applied to both generated spectra. For both examples the algorithm was run for 10000 iterations with a starting peptide of $GTMSGRSQ$, which was obtained from the results from PepNovo.

Figure 6.4 shows the plot of the simulated spectrum with minimal noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true peptide when using minimal noise is $TGMSNVSK$ with a log posterior density of 12.50 (up to a constant). Table 6.2 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum with minimal noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is the top estimated peptide, which ensures our method is performing substantially well.

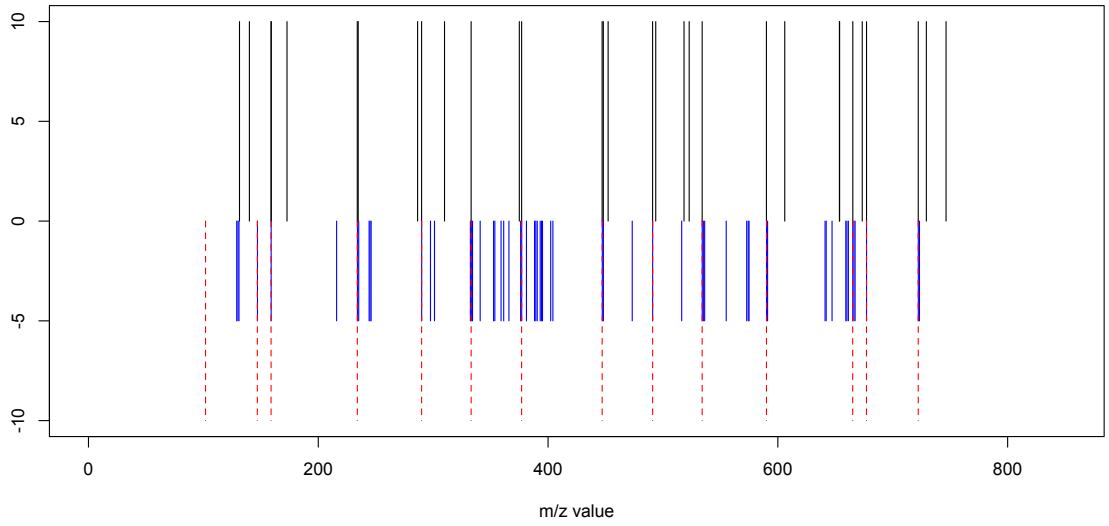

Figure 6.4 Simulated spectrum with minimal noise plotted against the observed and theoretical spectra when using the Laplace noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.2 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, and log κ_1 and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum with minimal noise.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
TGMSNVSK	12.50	45.39	-9.88	-27.45	5.22	-0.78
TGMSAAASK	10.76	41.15	-7.26	-27.09	4.87	-0.90
TGMGTAGTK	6.04	41.02	-9.95	-27.41	3.01	-0.64
TGMGTAASK	5.32	39.66	-9.23	-27.45	2.98	-0.60
TGMSAATGK	3.69	37.03	-8.51	-26.53	2.59	-0.88

Figure 6.5 shows the plot of the simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for

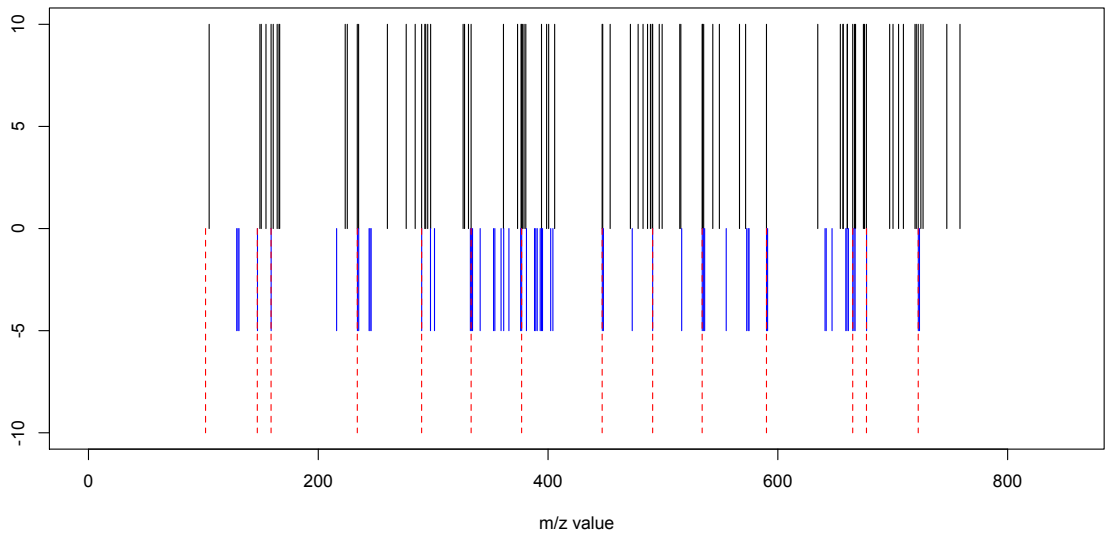

Figure 6.5 Simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum when using the Laplace noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

the true peptide when using substantial noise is $TGMSAAASK$ with a log posterior density of 40.36 (up to a constant). Table 6.3 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum with substantial noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Although the spectrum has more noise, our method was still able to identify the true peptide as being among the best choices.

Table 6.3 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, and log κ_1 and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum with substantial noise.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
TGMSAAASK	40.36	70.38	-8.15	-27.45	6.01	-0.74
TGMSGGVSK	39.60	71.46	-8.68	-28.31	5.90	-0.77
TGMSGVGSK	38.66	70.84	-8.67	-28.44	5.81	-0.88
SAMSNVSK	35.45	66.08	-9.62	-25.66	5.74	-1.09
TGMSNVSK	33.22	63.68	-9.56	-25.62	5.64	-0.92
TGMSNVSAG	27.10	63.40	-9.19	-31.85	5.32	-0.63
ASMSNVSK	21.70	52.06	-9.07	-25.62	5.01	-0.69
TGMSNGEK	21.40	48.65	-6.17	-25.62	4.98	-0.74
SAMSNVSQ	20.48	51.58	-6.40	-28.33	4.76	-1.13

Example 6.2

Here we set the parameters to be the same as in Example 6.1. We now generate a spectrum for a peptide with a longer amino acid sequence. The generated spectrum is based on the peptide $YHFEQSTVTSQPAR$ whose observed spectrum contains m/z values that range from 235 to 1634 Da. The total number of true peaks is $s = 26$ and so $s_1 = 3$, $s_2 = 20$, and $s_3 = 3$ using boundary section proportions 0.1. Figure 6.6 shows the plot of the simulated data versus the observed spectrum and theoretical spectrum. The simulated spectrum is above the zero axis. The observed spectrum is plotted below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines. One can see the simulated spectrum mimics the observed spectrum reasonably well.

After the spectrum is simulated, we then applied our method to the simulated spectrum. The starting peptide is $HYFETDQATSKPVK$, which was obtained from the results from PepNovo when applied to the real data. The algorithm was run for 10000 iterations and the best estimated peptide for the true peptide is $YHFEQSTVTSQPAR$ with a log posterior density of 88.52 (up to a constant). Table 6.4 shows the top estimated peptides for the $YHFEQSTVTSQPAR$ example using the simulated spectrum along with their

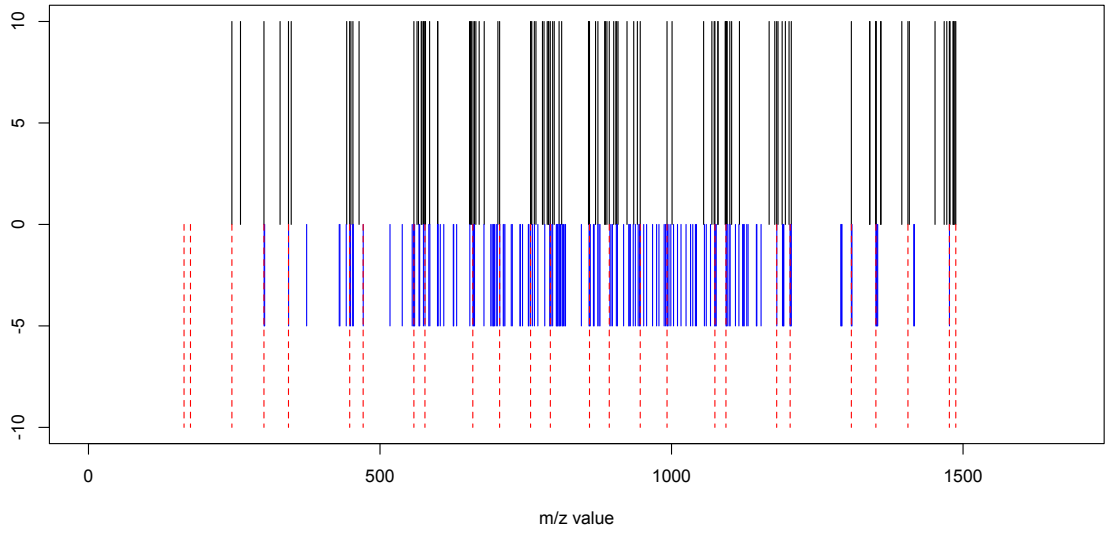

Figure 6.6 Simulated spectrum plotted against the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.4 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
YHFEQSTVTSQPAR	88.52	141.08	-13.57	-44.63	5.44	0.19
YHFEQSTVTNTPVQ	86.87	145.87	-17.28	-47.56	5.67	0.16
GDKFEQSTVTSQPAR	85.00	147.05	-18.65	-49.15	5.62	0.14
YHFEQSTVTNTPAR	84.01	134.36	-12.12	-43.85	5.53	0.09
YHFEKSTVTSQPAR	83.47	140.44	-14.70	-48.14	5.76	0.12
YHFEQSTVTSKPAR	70.50	126.80	-15.02	-46.89	5.58	0.03
YHFEKSTVTSKPAR	59.67	118.56	-13.70	-50.41	5.00	0.22
YHFAWSTVTSKPAR	43.31	102.52	-15.31	-49.07	5.07	0.10
YHFAWSTVTSQPAR	41.81	102.71	-18.84	-46.80	4.63	0.11

corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. As in Example 6.1, we see that the true peptide (highlighted in bold) is among the top estimated peptides. Here the best estimated peptide is the true peptide.

Once again, to how see our method is performing, we simulate a spectrum with minimal noise and one with substantial noise for a peptide with a longer amino acid sequence. To produce a generated spectrum with minimal noise, we decrease the values of b and as in Example 6.1, we set the values to be $b_1 = b_2 = b_3 = 2$. To produce a generated spectrum with substantial noise for a peptide with a longer amino acid sequence, we increase the values of b and as in Example 6.1, we set the values to be $b_1 = b_2 = b_3 = 10$. Our method was then applied to both generated spectra. For both examples the algorithm was run for 10000 iterations with a starting peptide of $HYFETDQATSKPVK$, which was obtained from the results from PepNovo.

Figure 6.7 show the plot of the simulated spectrum with minimal noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true

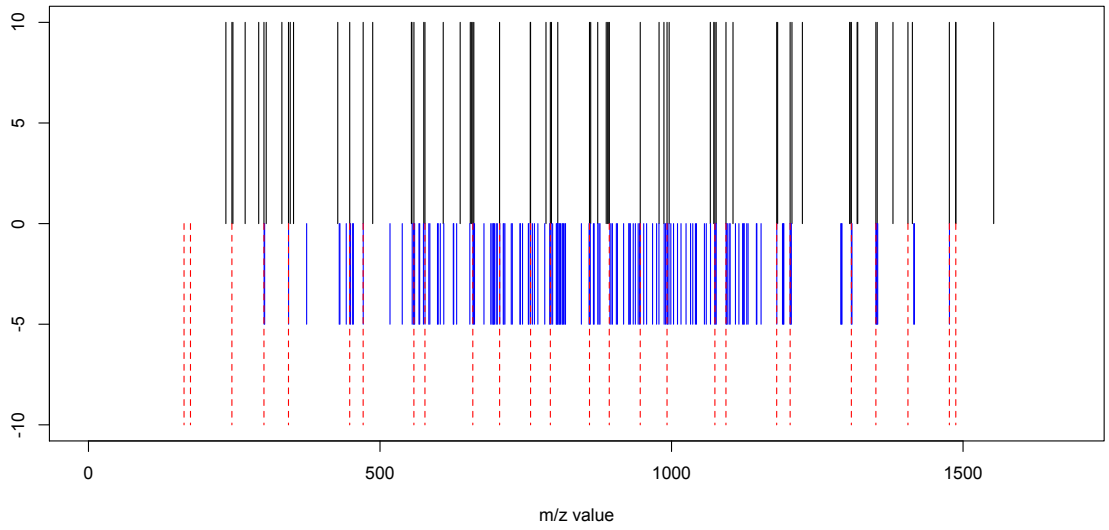

Figure 6.7 Simulated spectrum with minimal noise plotted against the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

peptide when using minimal noise is $YHFEQSTVTSQPAR$ with a log posterior density of 6.21 (up to a constant). Table 6.5 shows the top estimated peptides for the $YHFEQSTVTSQPAR$

Table 6.5 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum with minimal noise.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
YHFEQSTVTSQPAR	6.21	55.98	-10.29	-47.30	5.99	0.22
YHFSEGSSVVSQPAR	4.40	62.07	-16.64	-47.08	5.86	0.19
YHFSEKTVTSQPAR	3.60	55.12	-10.29	-47.30	5.82	0.25
YHFSEGATTVSQPAR	2.45	60.22	-16.64	-47.08	5.79	0.16
NVSFSEGSSVVSQPAR	-9.66	56.62	-22.93	-49.16	5.63	0.18
AWSVMAAGAGGFPRGES	-22.94	50.69	-14.88	-64.42	5.55	0.12
YHFSEGATVTSQPAR	-35.23	21.40	-15.42	-46.53	5.24	0.09
YNLGSENSPDCKVPQ	-39.69	22.63	-14.04	-53.50	5.08	0.13
VTTFGGNLVMAAELID	-41.07	22.13	-13.41	-54.85	4.88	0.18
LIAFFNGGGATCHEVD	-41.45	29.23	-16.33	-59.10	4.72	0.04

example using the simulated spectrum with minimal noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is the top estimated peptide as in Example 6.1 ensuring our method is performing substantially accurately.

Figure 6.8 shows the plot of the simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true peptide when using substantial noise is $YHFEQSTVTQSPLN$ with a log posterior density of 180.51 (up to a constant). Table 6.6 shows the top estimated peptides for the $YHFEQSTVTSQPAR$ example using the simulated spectrum with substantial noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. Here the true peptide was not identified in the top estimated peptides but there was more noise added into the spectrum, causing our method not to identify it as one of the top choices.

6.2 Poisson Noise Structure

A counting process is a stochastic process $\{N(t), t \geq 0\}$ where $\{N(t)\}$ is the total number of events that have occurred in a specified time interval, $[0, t]$. A Poisson process is a

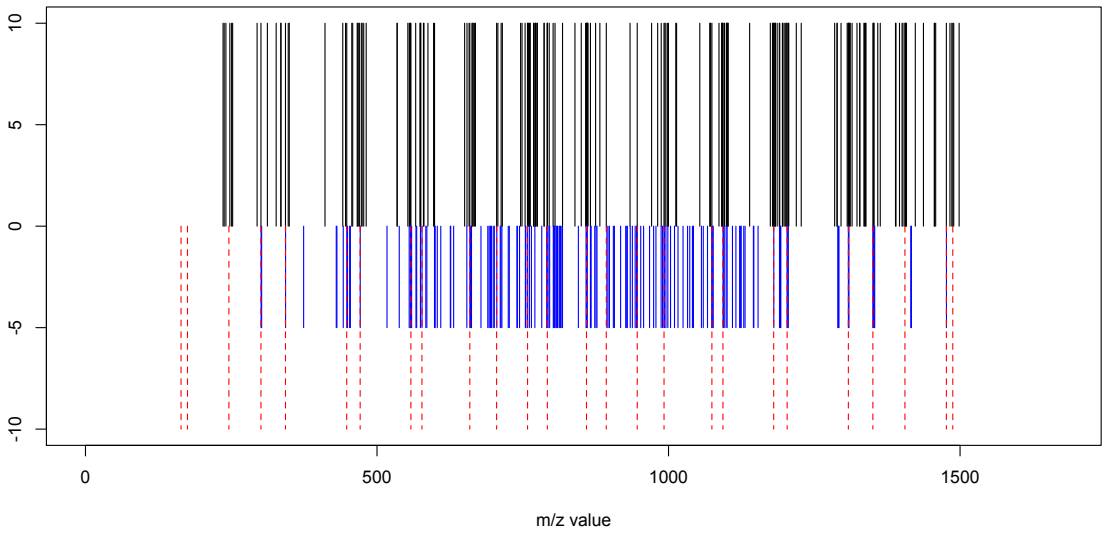

Figure 6.8 Simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.6 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum with substantial noise.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
YHFEQSTVTQSPLN	180.51	241.92	-18.71	-49.14	5.75	0.68
YHFEKSTVTQSPAR	163.42	220.25	-14.52	-48.20	5.16	0.73
YHFEKSTVTSKPAR	157.52	218.14	-16.35	-50.41	5.45	0.69
YHFEQSTVTQSPAR	154.39	206.48	-13.09	-44.69	5.02	0.67
YHFEQSTTWQPAR	133.61	185.51	-14.11	-43.13	4.65	0.68
YHFEKSTTWQPAR	128.38	185.13	-15.42	-46.64	4.67	0.64
YHFEKSTVTSQPAR	122.15	177.13	-11.86	-48.14	4.46	0.57
YHFEKSTVTSPQAR	120.26	181.71	-16.35	-50.41	4.82	0.48
YHFEQSTSISQPAR	116.96	169.27	-12.00	-45.25	4.39	0.55

type of counting process that has independent increments; that is, the number of events in the process occurring in nonoverlapping intervals are independent random variables. The time between pair of successive events follows an exponential distribution with rate parameter θ (Doob, 1953; Raftery and Akman, 1986). By using a Poisson process, we can generate noise peaks that are independent of the signal peak locations, and thus we explore generating the noise peaks by using a Poisson process.

As with the Laplace noise structure, the spectrum is split into three sections. To determine the number of noise peaks needed for each section, the total number of m/z values, denoted as q, that have intensity values above a specific threshold is first found from the observed spectrum of the true peptide. Then q is split into three values, (q_1, q_2, and q_3), where these values will determine the number of noise peaks needed for each section. Due to the processing of the spectrum by the mass spectrometer, the first section of the spectrum will have the fewest noise peaks and the middle section will have the most noise peaks. To illustrate how to find the number of peaks needed for each section, consider a peptide whose spectrum contains 100 m/z values. For the first section of the spectrum, there will be $q_1 = 100 \times .25 = 25$ noise peaks that are generated. The third section of the spectrum will have $q_3 = 100 \times .25 = 25$ noise peaks. The middle section will then have $q_2 = 100 - 25 - 25 = 50$ noise peaks. The proportions 0.25 and 0.75 for each boundary section were chosen based upon extensive numerical experimentation. To obtain the locations for the noise peaks for each section, the cumulative sum of randomly generated values from an exponential distribution, shifted by a specified value c, is found by the following algorithm:

1. Initialize $t = 0$.
2. Generate $x \sim Exp(\theta)$
3. Set $t = t + x$.
4. Store t in $\boldsymbol{t}$.

5. Repeat q_i times.

6. Compute $\boldsymbol{t} + c$,

where $\boldsymbol{t}$ is the vector of noise peaks for section i for $i = 1, 2, 3$. In order for the generated noise peaks to have m/z values in the same range as the observed spectrum, we must set an initial value to be added to the Poisson process. Recall the amino acid G has the smallest mass of 57.0215 Da. Let max_{mz} be the largest m/z value for the observed spectrum. For the first section, the value of c (denoted as c_1) is found by $c_1 = 57 + (max_{mz} - 57) \times 0.1$. The value for c for the middle section (denoted as c_2) is found by $c_2 = c_1 + (max_{mz} - 57) \times 0.2$ and for the last section c (denoted as c_3) is found by $c_3 = c_1 + c_2 + (max_{mz} - 57) \times 0.3$. The proportions $0.1, 0.2.$ and 0.3 are chosen after experimentation. To demonstrate how to find c, consider a peptide whose maximum m/z value is 1150. The values of c for each section would be the following: $c_1 = 57 + (1150 - 57) \times 0.1 = 166$, $c_2 = 166 + (1150 - 57) \times 0.2 = 385$, and $c_3 = 166 + 385 + (1150 - 57) \times 0.30 = 879$.

The full algorithm for obtaining the signal and noise peaks using a Poisson process with a parameter θ is defined as

1. Determine the total number of true peaks, s, and compute $\boldsymbol{\tau}$ by finding the b and y ions, based on the given peptide.

2. Simulate signal peaks from a density $f(x_i) \propto e^{-\kappa_1 |x_i - \tau_i|}$ for $i = 1, \ldots, s$.

3. Use an indicator function λ_i with probability function $P(\boldsymbol{\lambda}) = \prod_{i=1}^{p} P(\lambda_i^b, \lambda_i^y)$, where $p = s - 1$ is the total number of b ions (or, equivalently, the number of y ions) to determine the presence or absence of each signal peak.

4. For each of the three sections in the spectrum, compute the number of noise peaks for each section of the spectrum.

5. Simulate noise peaks from the algorithm described above.

When using a Poisson process to simulate the location of the noise peaks, we need to choose the value of the fixed parameter θ. Figure 6.9 shows the plot of a generated spectrum versus the observed spectrum and theoretical spectrum for a given peptide using different values of θ. This illustrates how the choice of θ affects the location of the generated noise peaks. The choice of θ for Figure 6.9 (a) is $\theta = 1/15$, $\theta = 1.0$ for Figure 6.9 (b), and $\theta = 1/30$ for Figure 6.9 (c). Increasing θ causes the location of the generated noise peaks to create clusters of tightly spaced noise peaks. Decreasing the value of θ causes there to be fewer noise peaks in the generated spectrum and thus it does not imitate the observed spectrum as well.

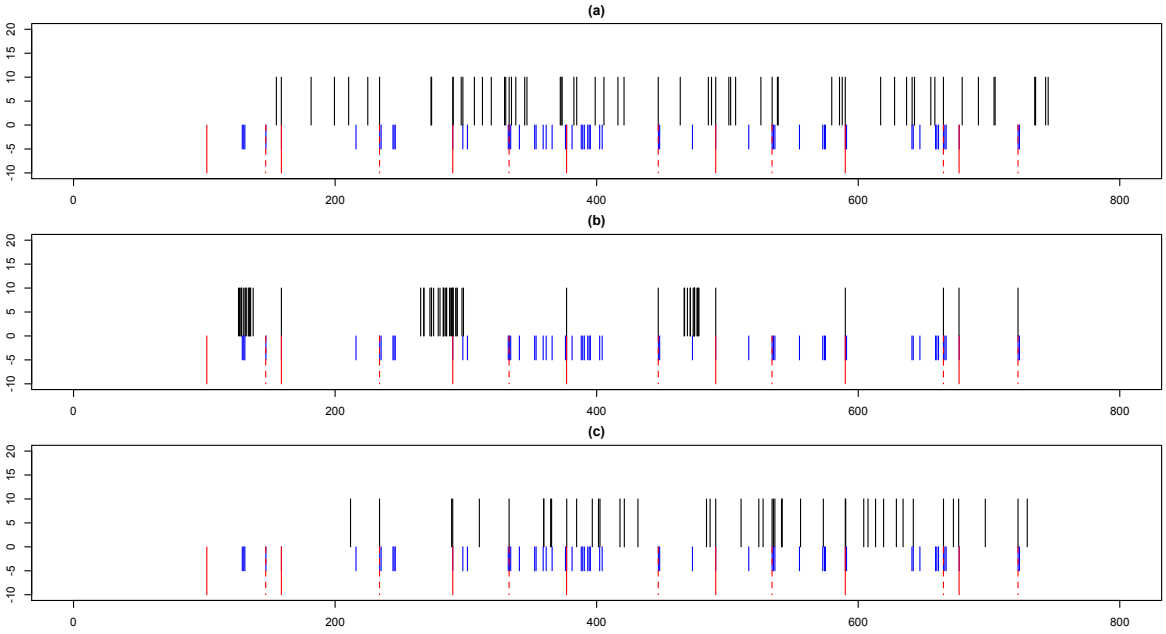

Figure 6.9 Simulated spectrum plotted against the observed spectrum and theoretical spectrum. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines. Values for θ: (a) $\theta = 1/15$, (b) $\theta = 1.0$, (c) $\theta = 1/30$.

Example 6.3

The parameters must first be specified. We set $\kappa_1 = 50$ and $\theta = 1/15$. As in Example 6.1 and 6.1, we set the first elements of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ to be $p_{b1} = 0.05$ and $p_{y_1} = 0.10$ and all other p_{bi} and p_{y_i} to equal 0.80 for $i = 2, \ldots, p$.

As in Example 6.1, we first generate a peptide with a short amino acid sequence. We use the same peptide as in Example 6.2, $TGMSNVSK$. Recall the observed spectrum contains m/z values that range from 123 to 749 Da. The total number of true peaks is $s = 14$ and so $s_1 = 1$, $s_2 = 12$, and $s_3 = 1$. The total number of m/z values in the observed spectrum with intensity values above a threshold is $q = 75$ and so $q_1 = 19$, $q_2 = 37$, and $q_3 = 19$. Figure 6.10 shows the plot of the simulated data versus the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is plotted below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines. One can see the simulated spectrum mimics the observed spectrum reasonably well.

After the spectrum is generated, our method described in Chapter 4 is then applied to the simulated spectrum. The starting peptide is $GTMSGRSQ$, which was obtained from the results from PepNovo when applied to the real data. The algorithm was run for 10000 iterations and the best estimated peptide for the true peptide is $SAMSNVSK$ with a log posterior density of 12.60 (up to a constant). Table 6.7 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum with Poisson noise structure along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Here it is the second best peptide but notice the log posterior for the best estimated and the log posterior for the true peptide is quite similar.

Once again, we need to ensure our algorithm is performing accurately, and so we sim-

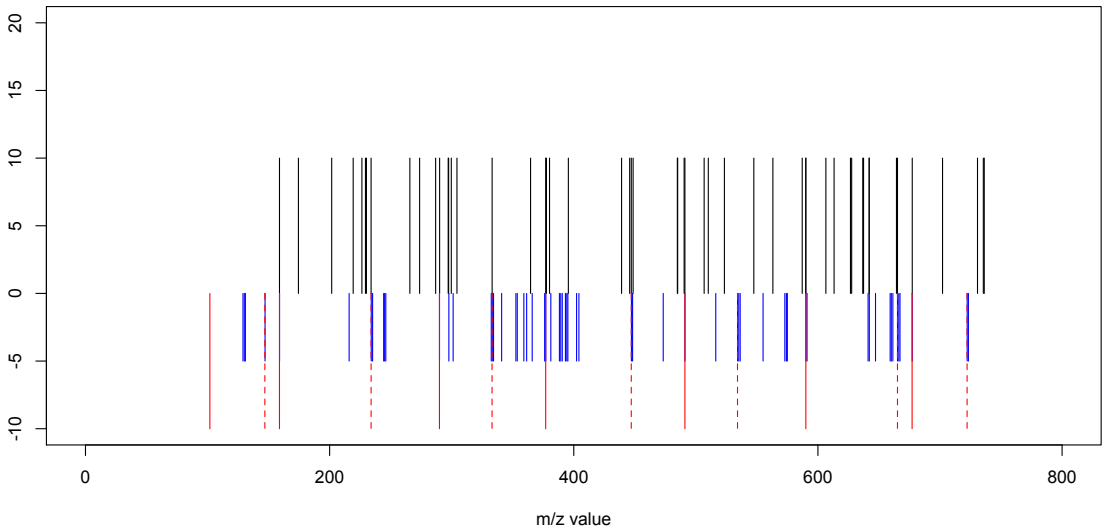

Figure 6.10 Simulated spectrum plotted against the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.7 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
SAMSNVSK	12.60	40.61	-7.14	-25.66	5.41	-0.62
TGMSNVSK	11.30	43.79	-11.66	-25.62	5.33	-0.55
GTMSNVSK	9.79	40.27	-8.84	-26.39	5.35	-0.60
SAFANVSK	9.30	37.84	-8.20	-24.92	5.13	-0.54
SAAFNVSK	3.29	28.90	-6.18	-23.85	5.14	-0.72
SAFANVNT	3.11	35.72	-8.53	-28.61	5.12	-0.60
SAFANVKS	0.69	36.95	-7.88	-32.84	5.06	-0.60
SAFANVSQ	-6.22	25.48	-8.16	-27.59	4.88	-0.83
SAMSNVSGA	-40.97	-9.21	-7.63	-25.66	2.59	-1.06
TGMSNVSGA	-61.57	-21.87	-9.56	-31.08	2.11	-1.16

ulate two spectra; one with minimal noise and one with substantial noise. To decrease the number of noise peaks, we decrease the values of q_1, q_2, and q_3 by 15 and to increase the number of noise peaks in the generated spectrum, we increase the values of q_1, q_2, and q_3 by 15. Our method was then applied to both generated spectra. For both examples the algorithm was run for 10000 iterations with a starting peptide of $GTMSGRSQ$, which was obtained from the results from PepNovo.

Figure 6.11 shows the plot of the simulated spectrum with minimal noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true

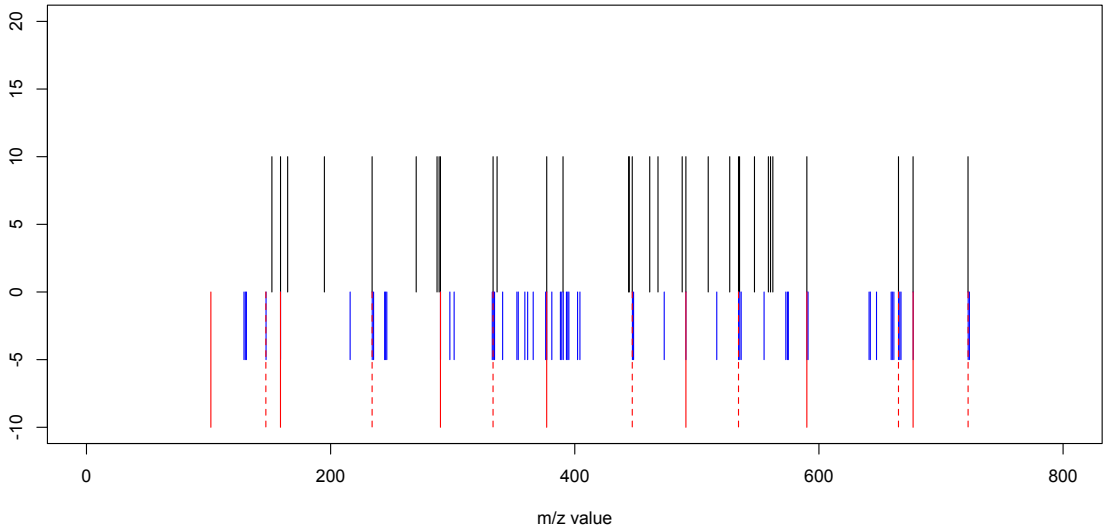

Figure 6.11 Simulated spectrum with minimal noise plotted against the observed and theoretical spectra when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

peptide when using minimal noise is $TGMSNVSK$ with a log posterior density of -2.03 (up to a constant). Table 6.8 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum with minimal noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is the

Table 6.8 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, and log κ_1 and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum with minimal noise.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
TGMSNVSK	-2.03	25.19	-6.57	-25.62	5.78	-0.81
TGMSRGSK	-24.68	8.33	-6.50	-31.16	5.33	-0.67
SVFNKTQ	-31.30	-1.63	-5.75	-28.69	5.22	-0.45
TGMSRGSQ	-36.43	0.58	-8.07	-33.83	5.47	-0.58
TFQNVSGA	-37.00	-4.01	-9.01	-28.40	5.19	-0.76
SVFNKSAA	-50.01	-14.80	-8.04	-31.34	4.96	-0.79
TGMGDKSQ	-50.85	-10.26	-14.22	-31.16	5.33	-0.54
TFQGGVSGA	-52.06	-16.93	-8.38	-31.25	5.12	-0.62
SFNEGGAAA	-53.32	-19.94	-8.13	-29.22	4.24	-0.27
NSVAAHPQ	-54.21	-22.24	-7.99	-29.14	5.38	-0.22

top estimated peptide indicating our method is performing substantially accurately.

Figure 6.12 shows the plot of the simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true peptide when using substantial noise is $SAMSGGVSK$ with a log posterior density of 17.97 (up to a constant). Table 6.9 shows the top estimated peptides for the $TGMSNVSK$ example using the simulated spectrum with substantial noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Although the spectrum has more noise peaks, our method was still able to identify the true peptide as being among the best choices.

Example 6.4

For the second example, we keep the parameters the same as specified in Example 6.3. We consider a peptide with a longer amino acid structure. We generate a spectrum for the peptide $YHFEQSTVTSQPAR$, which is the same as in Example 6.2. Recall the

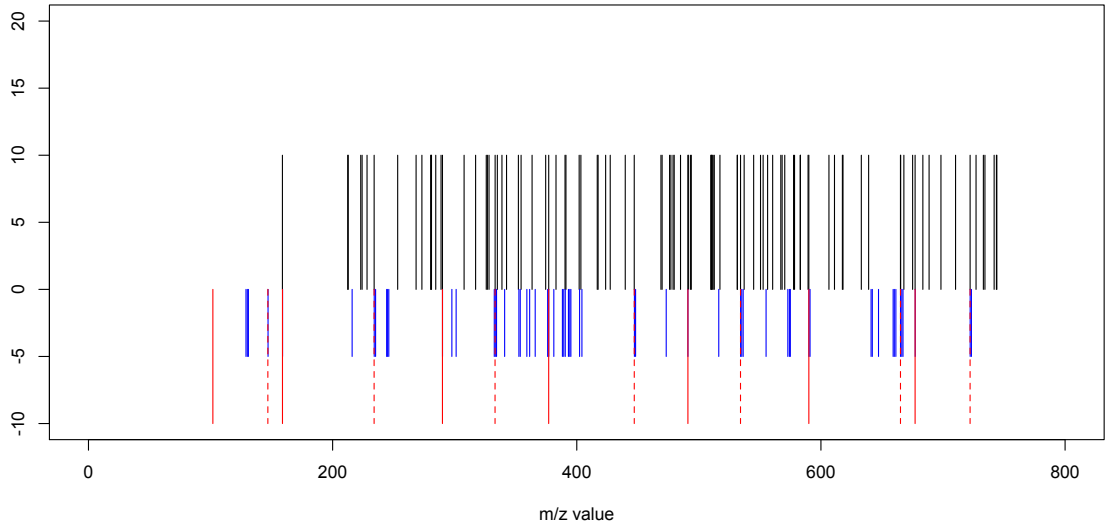

Figure 6.12 Simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.9 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $TGMSNVSK$ when using a simulated spectrum with substantial noise.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
SAMSGGVSK	17.97	49.44	-7.79	-28.35	5.01	-0.34
TGMSGGVSK	14.31	46.91	-8.81	-28.31	4.85	-0.34
TGMSNVSK	0.55	29.41	-7.56	-25.62	4.86	-0.55
TGMSNVSAG	-3.03	33.35	-8.72	-31.85	4.76	-0.57
TGSMNVSK	-3.33	25.64	-8.02	-25.15	4.79	-0.59
TGSMGGVSK	-3.44	30.09	-8.81	-28.31	4.22	-0.64
TGMSNVSGA	-6.45	30.13	-9.84	-31.04	4.98	-0.69
TMGSNVSK	-7.21	21.85	-7.56	-25.62	4.82	-0.70

observed spectrum contains m/z values that range from 235 to 1634 Da. The total number of true peaks is $s = 26$ and so $s_1 = 3$, $s_2 = 20$, and $s_3 = 3$. The total number of m/z values in the observed spectrum with intensity values above a threshold is $q = 157$ and so $q_1 = 39$, $q_2 = 79$, and $q_3 = 39$ (up to a constant).

Figure 6.13 shows the plot of the simulated data versus the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is plotted below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines. One can see the simulated spectrum mimics the observed spectrum reasonably well. After

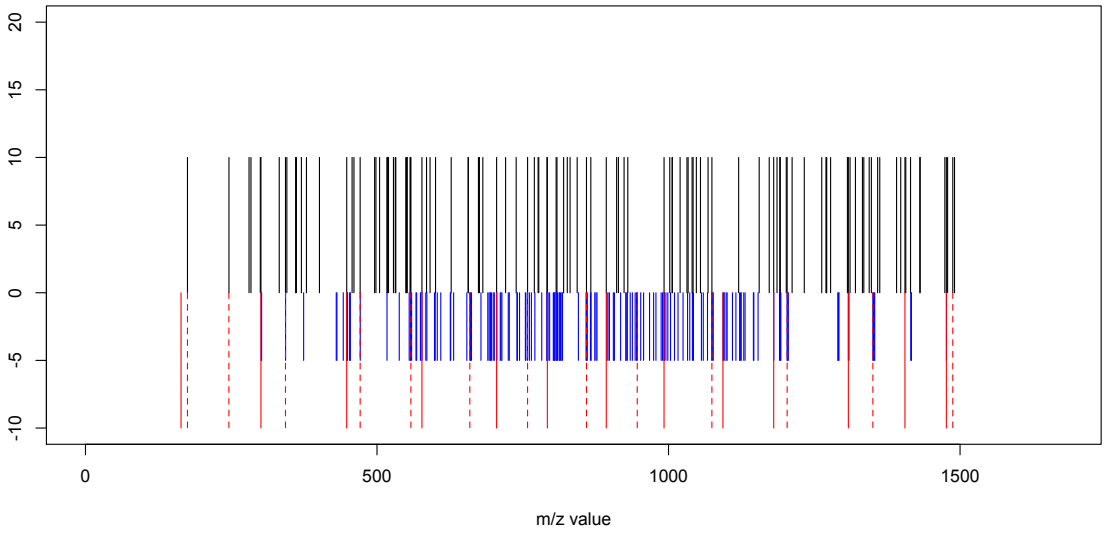

Figure 6.13 Simulated spectrum plotted against the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

the spectrum is simulated, our method is applied to the simulated spectrum. The starting peptide is $HYFETDQATSKPVK$, which was obtained from the results from PepNovo when applied to the real data. The algorithm was run for 10000 iterations and the best estimated peptide for the true peptide is $YHFGATMSVEGQPAVG$ with a log poste-

Table 6.10 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
YHFGATMSVEGQPAVG	10.69	76.27	-13.98	-55.72	4.04	0.09
YHFQTFAVEGQPAVG	8.55	74.36	-17.51	-52.15	3.76	0.10
YHFEQSTVTSQPAR	6.41	77.81	-20.77	-54.41	3.67	0.10
YHFGATMEGEGQPAVG	-3.28	67.19	-20.01	-53.99	3.40	0.11
YHFGAGMETEGQPAVG	-9.41	59.44	-18.46	-53.92	3.48	0.05
YHFGATSETEGQPAVG	-10.08	59.58	-19.57	-53.74	3.64	0.01
YHFGAGMETADQPAVG	-13.19	59.62	-22.10	-54.17	3.41	0.05
YHFGASTETEGQPAVG	-17.15	51.30	-17.65	-54.01	3.25	-0.03
YHFGAGMETADQPAR	-21.03	39.27	-16.27	-47.09	3.13	-0.06
YHFKTDTVTSQPAR	-30.34	32.64	-17.18	-48.78	3.00	-0.03
YHFGATFAVEGQPAVG	-47.16	18.95	-20.45	-48.14	2.67	-0.17

rior density of 10.69 (up to a constant). Table 6.10 shows the top estimated peptides for the $YHFEQSTVTSQPAR$ example using the simulated spectrum with Poisson noise structure along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Here it is the third best peptide.

Once again, we need to ensure our algorithm is performing accurately, and so we simulate two spectra: one with minimal noise and one with substantial noise. To decrease the number of noise peaks, we decrease the values of q_1, q_2, and q_3 by 15 and to increase the number of noise peaks in the generated spectrum, we increase the values of q_1, q_2, and q_3 by 15. Our method was then applied to both generated spectra. For both examples the algorithm was run for 10000 iterations with a starting peptide of $HYFETDQATSKPVK$, which was obtained from the results from PepNovo.

Figure 6.14 shows the plot of the simulated spectrum with minimal noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true peptide when using minimal noise is $YHFEQSTVTSPQAR$ with a log posterior density of 50.94 (up to a constant). Table 6.11 shows the top estimated peptides for the

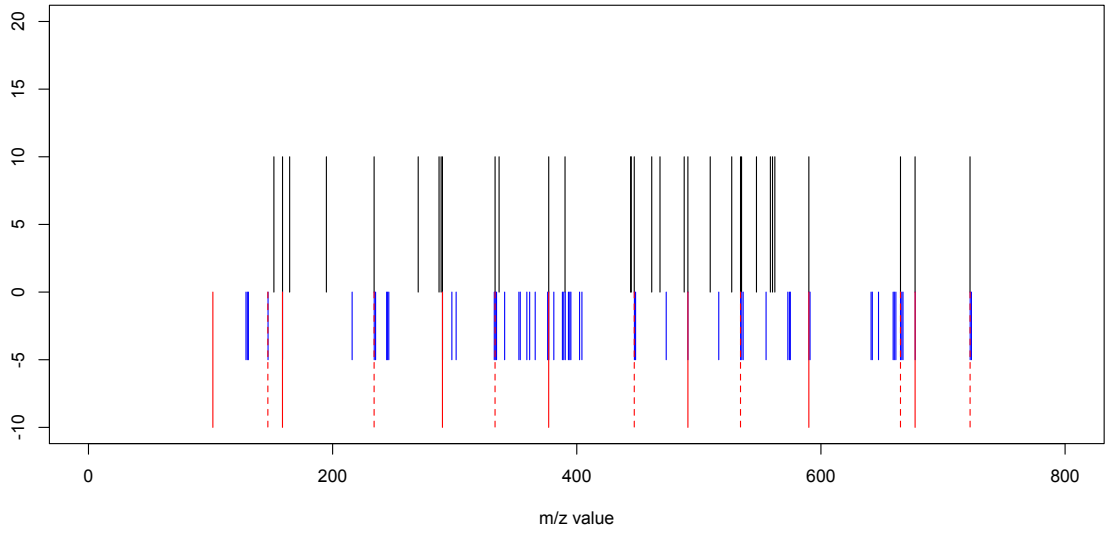

Figure 6.14 Simulated spectrum with minimal noise plotted against the observed and theoretical spectra when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

$YHFEQSTVTSQPAR$ example using the simulated spectrum with minimal noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide (highlighted in bold) is among the top estimated peptides. Here it is the second best estimated peptide, but do note the log posterior for true peptide is not that much smaller than the best estimated peptide, confirming that our method is performing adequately well.

Figure 6.15 shows the plot of the simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum. The best estimated peptide for the true peptide when using substantial noise is $YHFEQSTVTSAGPAR$ with a log posterior density of 141.50 (up to a constant). Table 6.12 shows the top estimated peptides for the $YHFEQSTVTSQPAR$ example using the simulated spectrum with substantial noise along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true

Table 6.11 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum with minimal noise.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
YHFEQSTVTSPQAR	50.94	106.71	-16.09	-45.21	5.01	0.52
YHFEQSTVTSQPAR	48.16	101.54	-14.20	-44.63	4.96	0.49
YHFEQSTVTSKPAR	44.39	100.61	-14.65	-46.89	4.88	0.44
YHFEQSTVTSKPNL	41.20	103.98	-16.84	-50.91	4.55	0.41
YHFEQSTPCSPAGAR	-9.05	58.66	-14.34	-58.55	4.78	0.39
YHFEQSTPCSAGPAR	-22.90	45.73	-15.31	-58.57	4.77	0.48
YHFEQSTPSSSGPAR	-42.21	20.86	-20.61	-47.18	4.40	0.32
YHFEQSPTSSSGPAR	-46.12	16.82	-16.76	-50.69	4.23	0.29
YHFEQSTPCSPQAR	-47.87	20.47	-14.34	-58.55	4.18	0.36
YHFEKSPTSSSGPAR	-68.16	0.13	-15.63	-57.43	4.33	0.45

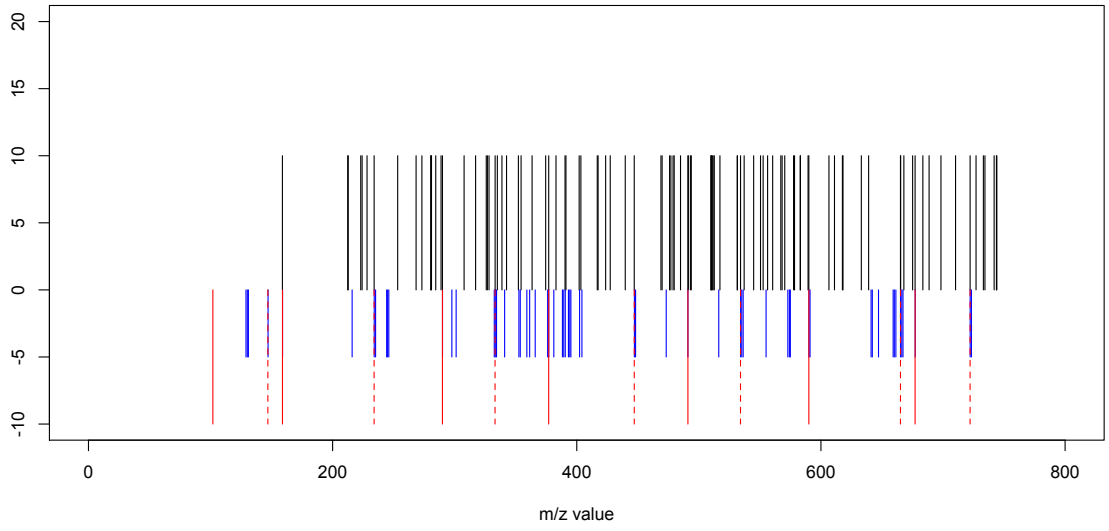

Figure 6.15 Simulated spectrum with substantial noise plotted against the observed spectrum and theoretical spectrum when using the Poisson noise structure. The simulated spectrum is plotted above the zero axis. The observed spectrum is below the zero axis with solid lines and the theoretical spectrum is plotted below the zero axis with dashed lines.

Table 6.12 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $YHFEQSTVTSQPAR$ when using a simulated spectrum with substantial noise.

Peptide	Log Posterior Densities	Log Likelihood	Log Cleavage Prior	Log Sequence Prior	Log κ_1 Prior	Log κ_2 Prior
YHFEQSTVTSAGPAR	141.50	194.83	-18.51	-47.21	5.75	0.45
YHFEAGSTVTSQPAR	140.62	190.67	-15.33	-47.04	5.70	0.46
YHFEQSTVTSGAPAR	140.22	192.03	-17.16	-47.31	5.92	0.40
YHFEAGSTVTSGAPAR	135.46	190.34	-17.10	-49.72	5.58	0.41
YHFEAGSTVTSKPAR	133.49	183.83	-13.50	-49.31	5.87	0.36
YHFEQSTVTSQPAR	128.28	173.90	-13.44	-44.63	5.86	0.37
YHFEQSTVMGQPAR	116.28	165.56	-15.99	-45.28	5.66	0.33

peptide (highlighted in bold) is among the top estimated peptides. Although the spectrum has more noise peaks, our method was still able to identify the true peptide as being among the best choices.

6.3 Comparison of Noise Structure

With moderate noise, both the methods perform equally well for both noise structures for peptides with both short and long amino acid sequences. When generating minimal noise, once again both noise structures performed equally well for peptides with short and long amino acid structures. With minimal noise, the true peptide was identified in all cases except with the Poisson noise structure for a peptide with a longer amino acid sequence, but the true peptide was estimated as the second best with a log posterior similar to the best estimated peptide.

For peptides with short and long amino acid sequences, using either noise structure, our method performed equally well for all levels of noise tried. The noise generated in the spectrum is not completely realistic since certain peak features like isotopic peak, adducts, and post translational modifications are not included in the noise and so the noise generated is somewhat artificial. Although the noise generated in the spectrum may not be realistic, the methods are promising. An advantage of using a Laplace noise structure is that our model would be an approximation to a generative model. Unlike a discriminative model,

a generative model allows one to generate samples from the joint distribution. Generative models are more flexible since they are full probabilistic models of all variables and can be used to simulate values of any variable in the model (Singla and Domingos, 2005). Note "our model" used is an approximation to the Laplace model.

CHAPTER 7

REAL DATA APPLICATION

Most peptides in the PNNL dataset described in Section 1.1 are of length 8 to 20 amino acids. Our data include some relatively longer peptides due to the type of equipment used to process the data. Recall the equipment used was a LTQ Orbitrap mass spectrometer, which is a hybrid machine composed of a linear ion trap mass spectrometer and the Orbitrap mass analyzer that uses a fast Fourier transform algorithm (Yates et al., 2009). The dataset contains 1,206 peptides with lengths ranging from 7 to 31 amino acids and an average length of 15.16. The data are doubly charged and the total mass for each peptide is given. The dataset contains a set of masses and corresponding intensities with an average intensity value of 50.7.

The data must first be pre-processed. We choose to remove the doubly charged parent ion from the dataset. A parent ion is the fragment ion generated in mass spectrometry before the ion is broken apart into further ions. The m/z value of the doubly-charged ion is $\frac{\sum_{i=1}^{K} m(p_i) + 1}{2}$. Therefore, we remove the peak at that m/z value. After extensive numerical experimentation, we found that using the 75th percentile to calculate the constant and moving threshold works well. This means we use the observed m/z values in the data that have corresponding observed intensity values above the threshold value in $\boldsymbol{T}$. The mass spectrometer is not always accurate and this can cause the ion fragments that are detected to be slightly shifted from their theoretical position. Therefore, we use a tolerance level of 0.5 Da. That is, we tolerate the ion peak locations up to ± 0.5 Da from their theoretical positions. We set the initial components of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ to be $p_{b1} = 0.05$ and $p_{y_1} = 0.10$. We set these probabilities low because the mass spectrometer rarely captures the first b

and y ion. We set all other p_{bi} and p_{y_i} to equal 0.80 for $i = 2, \ldots, p$. We must also specify the hyperparameters in the Gamma prior distribution for κ_1 and κ_2. After extensive numerical experimentation, the values of a_1, b_1, a_2, and b_2 were set to be 5.5, 0.1, 3, and 1/0.01, respectively. Figure 7.1 portrays the conditional posterior density function for κ_1 and Figure 7.2 portrays the prior density function for κ_2, given $S_1 = 0.16$ and $S_2 = 7.89$, which are fairly typical values of S_1 and S_2. Section 7.4 goes into more detail about the choice of the tuning parameters through exploratory analysis.

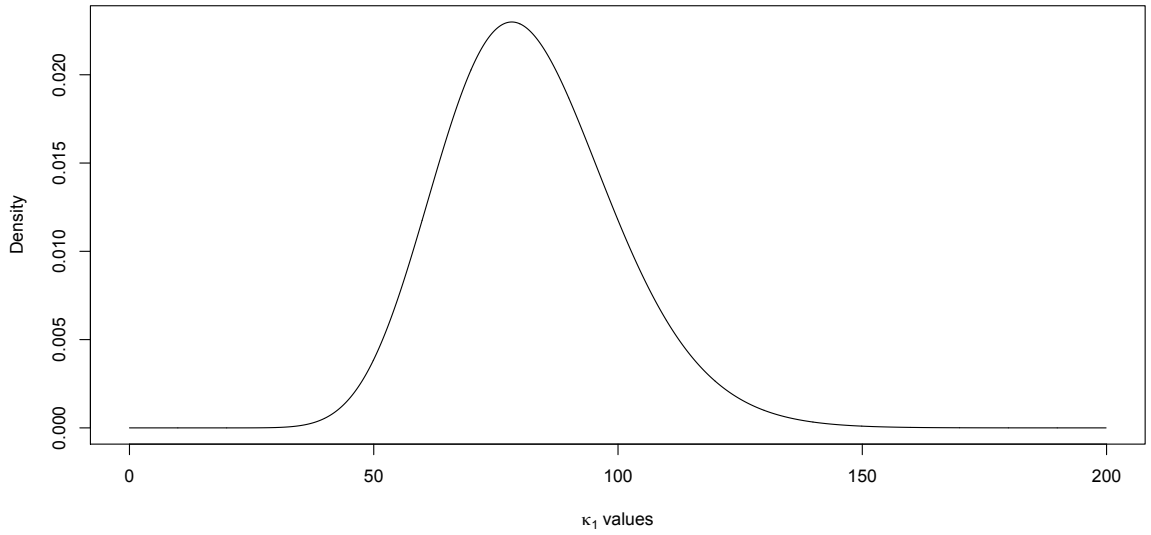

Figure 7.1 The conditional posterior density function for κ_1, given $S_1 = 0.16$ and $S_2 = 7.89$.

7.1 EXAMPLE 1

Consider the peptide $TGMSNVSK$. Figure 7.3 is a plot of the observed spectrum for the peptide $TGMSNVSK$. The theoretical spectrum is plotted below the zero axis. One can see that the theoretical spectrum aligns nicely with the observed spectrum. Note that there is quite a bit of noise in the center of the graph even after thresholding.

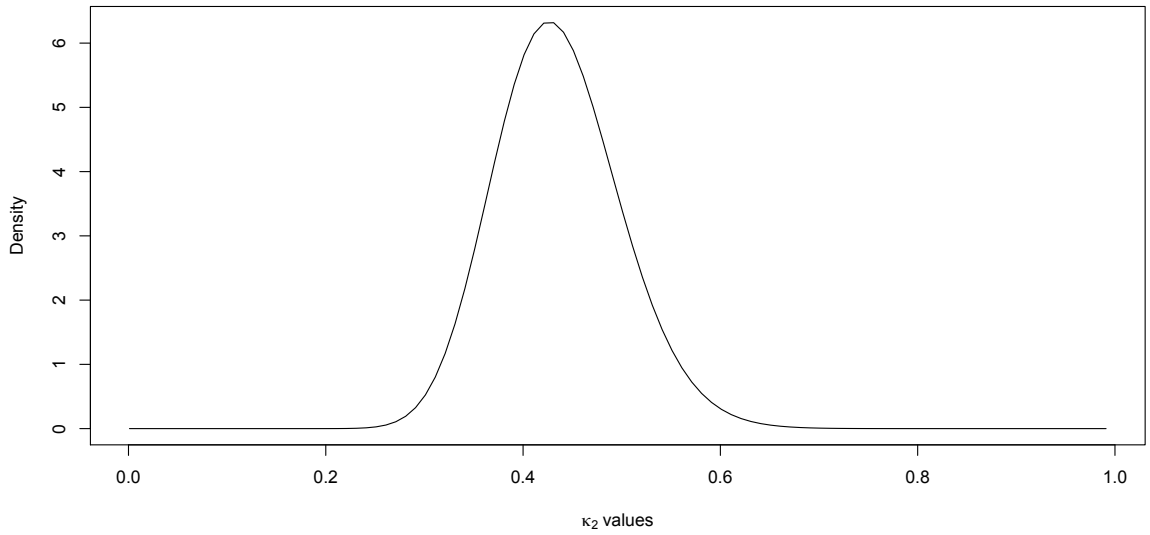

Figure 7.2 The conditional posterior density function for κ_2, given $S_1 = 0.16$ and $S_2 = 7.89$.

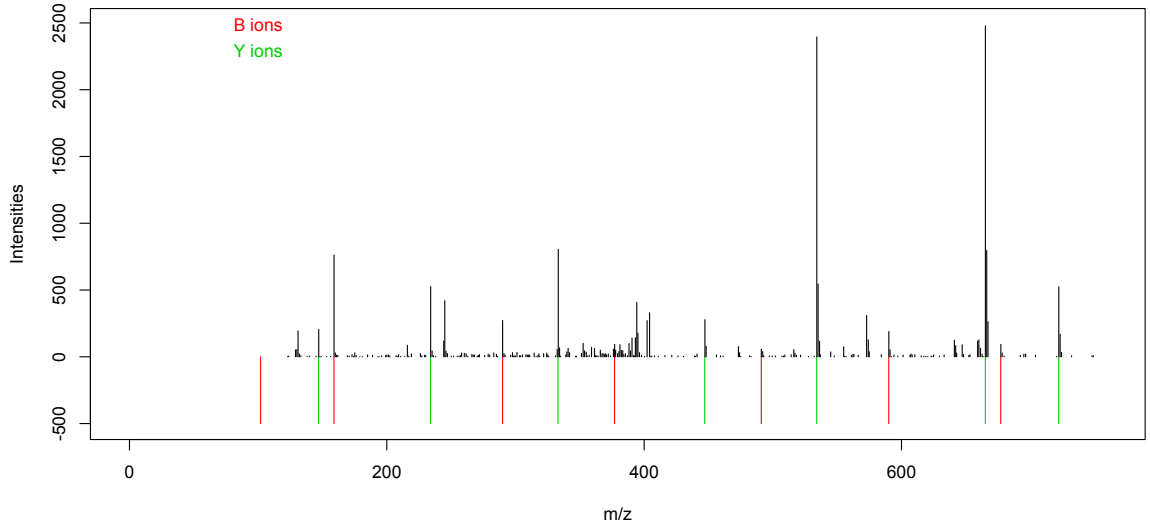

Figure 7.3 The observed spectrum plotted against the theoretical spectrum for the peptide $TGMSNVSK$. The theoretical spectrum is plotted below the zero axis and the observed spectrum is plotted above the zero axis.

Table 7.1 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $TGMSNVSK$ when using a random starting peptide.

Peptide	**Log Posterior Densities**
TMTISASL	-18.16
TMTISTGL	-18.23
TDDDSGTL	-20.70
TMVDSGTL	-24.46
TMTISGDV	-25.52
TMTISGTL	-28.18
TMVDSGIT	-29.37
TVVFSGTL	-29.77
TMTISGIT	-32.79
IDAFSGTL	-44.14

Using the initial iterative sub-algorithm, we obtain a starting peptide of $IEYGGGID$, which has a total mass that is within 0.5 Da of the weight of the true peptide. Running the algorithm for 10,000 iterations, we estimate the true peptide to be $TMTISASL$ with a log posterior density of -18.16 (up to a constant). Table 7.1 shows the top estimated peptides for the $TGMSNVSK$ example along with their corresponding estimated log posterior densities when the starting peptide is $IEYGGGID$. One can see the starting peptide is far from the true peptide and when only running the algorithm for 10,000 iterations, our method did not capture the true peptide. Starting from a completely random place is idealistic, but it may not be best in practice due to the fact there is such a large state space.

Now using the results from PepNovo, we obtain a starting peptide of $TGFAGGVSGA$, which has a total mass that is within 0.5 Da of the weight of the true peptide. Running the algorithm for 10,000 iterations, we obtain the best estimated for the true peptide to be $TGMSGGVSK$ with a log posterior density of 14.79 (up to a constant). Table 7.2 shows the top estimated peptides for the $TGMSNVSK$ example along with their corresponding estimated log posterior densities with the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide is among the top estimated peptides. Here it is the third best peptide. The log posterior for the first three best esti-

Table 7.2 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $TGMSNVSK$. The true peptide is in bold.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
TGMSGGVSK	14.79	49.16	-10.47	-28.28	4.67	-0.29
TGMSGGSVK	14.17	48.10	-10.19	-28.31	5.00	-0.43
TGMSNVSK	11.27	39.63	-8.92	-24.10	5.02	-0.37
TGAFGGSVK	-0.31	33.37	-10.36	-27.43	4.60	-0.50
TGAFGGWK	-8.30	24.63	-10.41	-26.68	4.60	-0.43
TGAFNWK	-11.00	14.06	-6.46	-22.83	4.72	-0.50
TGMSNWK	-18.39	10.00	-8.47	-24.01	4.72	-0.64
TGMSAKTK	-36.00	-1.57	-9.25	-28.20	3.84	-0.82
TGMANTTK	-36.07	-2.29	-7.93	-28.60	3.27	-0.53
TGMAKSTK	-37.25	-3.11	-7.93	-28.60	2.93	-0.55

mated peptides is fairly similar. In Figure 7.3 that we see there is additional noise in the center of the spectrum that causes the value of S_2 for the true peptide to be larger, and the difference in the log likelihood of the best estimated parameter and the true peptide can be seen in Table 7.2. Also, note the difference in the sequence prior for the best estimated peptide and the true peptide implying the amino acid sequence $TGMSGGVSK$ is more probable than the amino acid sequence $TGMSNVSK$. Figure 7.4 is the trace plot of the log posterior versus the number of iterations. One can see the chain has converged to its stationary distribution. Figure 7.5 is the trace plot of log κ_1 versus the number of iterations and Figure 7.6 is the trace plot of log κ_2 versus the number of iterations. We see the κ_1 and κ_2 samples converge to their stationary distribution.

To ensure our method is obtaining similar results for various starting peptides, we look at results from using different starting peptides that we obtain from PepNovo. Consider the starting peptides: $SAMYHSK$, $TGAFGRSK$, and $GTFANEGK$. Table 7.3 shows the top estimated peptides along with their corresponding log posterior densities for the above starting peptide values. One can see that the results are similar and that the true peptide (highlighted in bold in the table) is captured and is in the top 10 of the estimated peptides

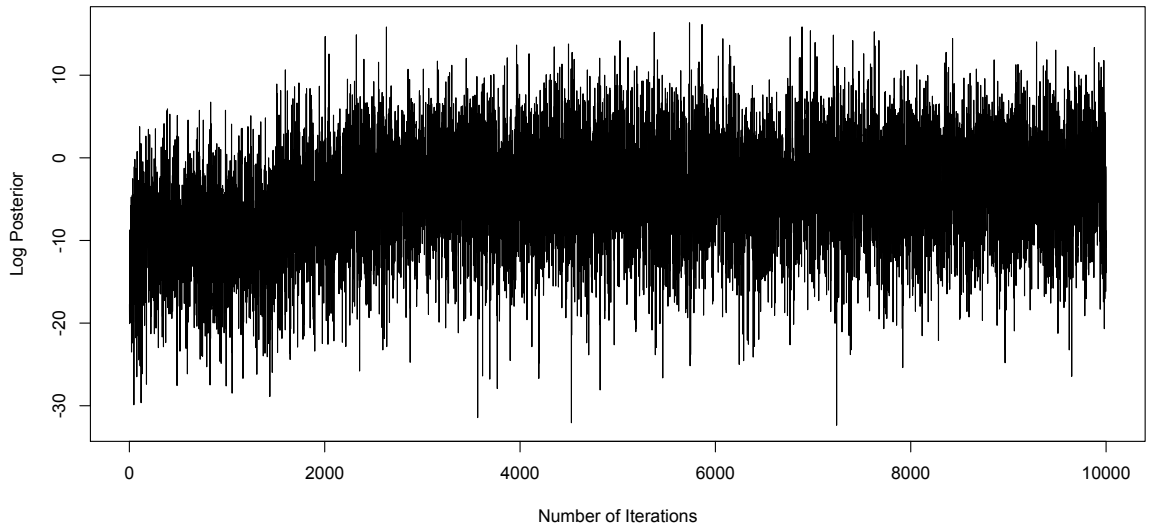

Figure 7.4 Trace plot of the log posterior for the peptide $TGMSNVSK$.

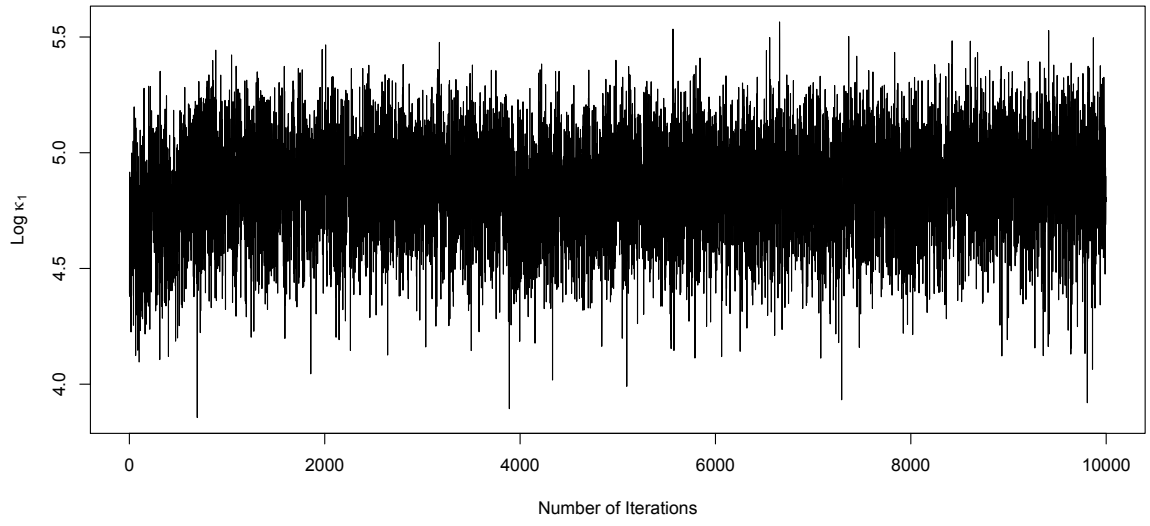

Figure 7.5 Trace plot of log κ_1 for the peptide $TGMSNVSK$.

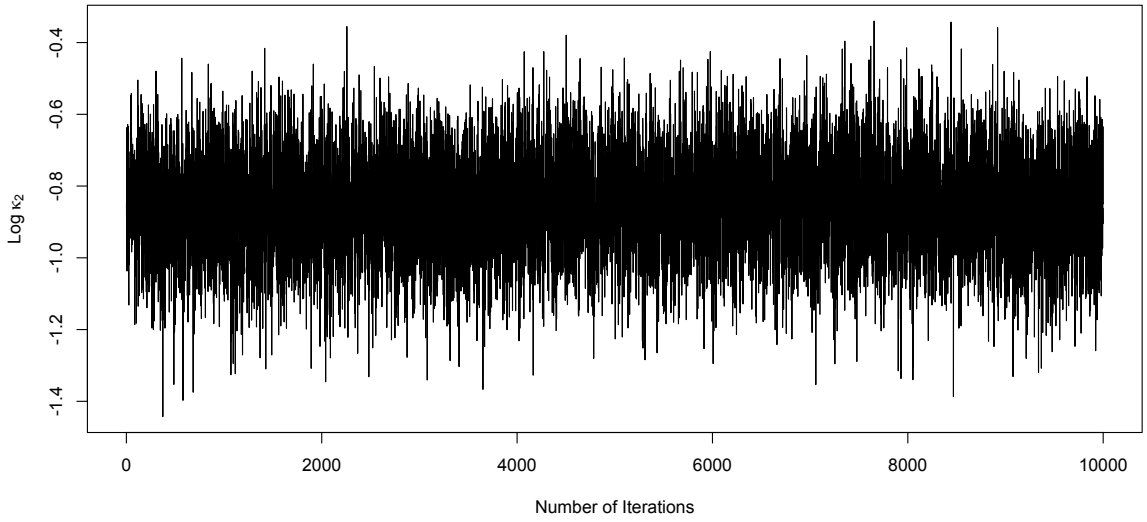

Figure 7.6 Trace plot of log κ_2 for the peptide $TGMSNVSK$.

Table 7.3 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $TGMSNVSK$ for three different starting peptides. The true peptide is in bold.

Starting Peptide					
SAMYHSK		**TGAFGRSK**		**GTFANEGK**	
Peptide	Log Posterior	Peptide	Log Posterior	Peptide	Log Posterior
TSTGGSVSK	2.40	TGSSSGLSK	0.32	TGMSGGVSGA	4.47
TSTSGGVSK	-1.01	TGSSSGISK	-0.15	TGSTSGVSGA	3.23
TGMSNSVK	-12.65	SAMSNVSK	-6.19	TGMGSGVSGA	0.91
TSTSNVSK	-14.86	SAMSNADK	-6.92	TGMSGGVSAG	-.00039
AAFSNVSK	-30.98	SANFAVSK	-7.13	TGMSGGVSK	-4.59
AAFKDGSK	-41.49	**TGMSNSVK**	-7.84	ASMSGGVSK	-9.50
AAFGWGSK	-46.46	TGFANVSK	-9.91	TGMSNVSK	-13.39
AAFYHSK	-47.15	SASSSAVSK	-10.04	**TGMSNSVK**	-15.51
SAMYHSK	-49.33	TGSSSAVSK	-10.54	SAMSNSVK	-28.03
SDSYHSK	-57.06	TGMSNVSQ	-13.64	SAFANSVK	-40.82

in all three cases. The log posterior densities are also similar in the three cases.

7.2 EXAMPLE 2

Consider a second peptide, $DLVESAPAALK$. Figure 7.7 is a plot of the observed spectrum for the peptide $DLVESAPAALK$. The theoretical spectrum is plotted below the zero axis. One can see that the theoretical spectrum aligns nicely with the observed spectrum.

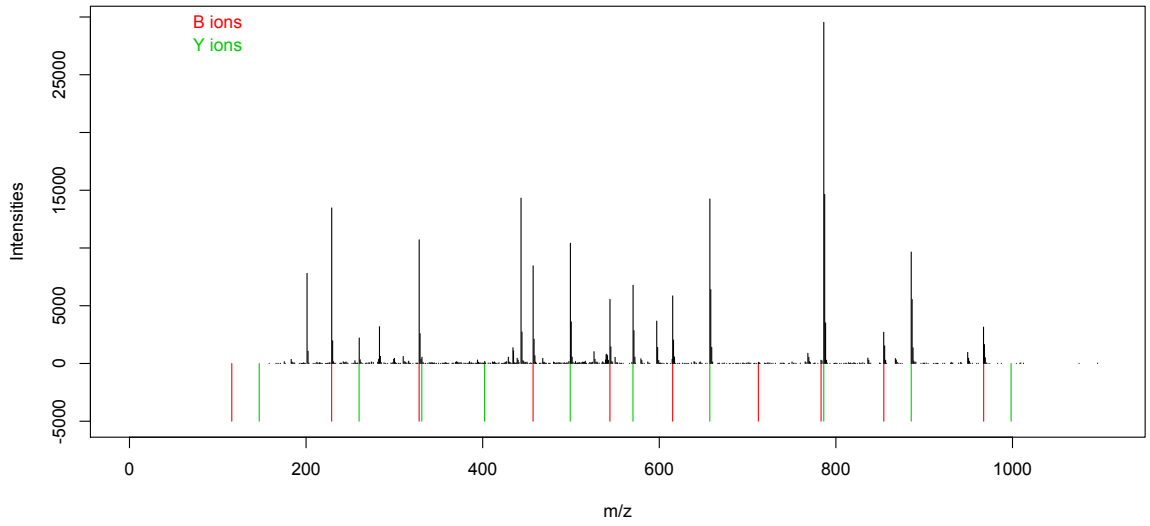

Figure 7.7 The observed spectrum plotted against the theoretical spectrum for the peptide $DLVESAPAALK$. The theoretical spectrum is plotted below the zero axis and the observed spectrum is plotted above the zero axis.

Using the initial iterative sub-algorithm, we obtain a starting peptide of $HMRAMPDQQ$, which has a total mass that is within 0.5 Da of the weight of the true peptide. Running the algorithm for 10,000 iterations, we estimate the true peptide to be $IAAAAGAAGGAANK$ with a log posterior density of 18.08 (up to a constant). Table 7.4 shows the top estimated peptides for the $DLVESAPAALK$ example along with their corresponding estimated log posterior densities when the starting peptide is $HMRAMPDQQ$. As in Example 1 in Section 7.1, the starting peptide is far from the true peptide and when only running the algorithm for 10,000 iterations, our method did not place the true peptide among the best

Table 7.4 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ when using a random starting peptide.

Peptide	Log Posterior Densities
IAAAAGAAGGAANK	18.08
IIVEAANAGQK	3.76
IAAAAGAAGGAGQK	2.36
INVKAAGGAGQK	1.78
IAAAKAAGGAGQK	-2.82
IIVEAAGGAGQK	-5.60
IIVEQQAGQK	-12.31
IIVEQQQQK	-18.88
IIVEPWTQK	-23.09
IIVVANQTQK	-24.78

estimates.

Here we obtain an initial peptide of $DLVESYFLK$ from the PepNovo results. Running the algorithm for 10,000 iterations, we obtain the estimated true peptide to be $IDVESAPAALK$ with a log posterior density of 65.92 (up to a constant). Table 7.5 shows the top estimated peptides for the $DLVESAPAALK$ example along with their corresponding estimated log posterior densities and the breakdown of the log posterior. The log likelihood and log priors are also given. We see that the true peptide is in the list of the top estimated peptides. Here it is the second best peptide. Note that the log posterior for the first three best estimated peptides is fairly similar. In Figure 7.7 we see there is additional noise in the center of the spectrum that causes the value of S_2 for the true peptide to be larger. Notice in Table 7.5 the difference in the log likelihood of the best estimated parameter and the true peptide, causing our method not to choose the true peptide as the best estimated peptide. Figure 7.8 is the trace plot of the log posterior versus the number of iterations. One can see the chain has converged to its stationary distribution. Figure 7.9 is the trace plot of log κ_1 versus the number of iterations and Figure 7.10 is the trace plot of log κ_2 versus the number of iterations, respectively. Once again, we see the κ_1 and κ_2 samples converge to their stationary distribution.

Table 7.5 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities, log likelihoood, log cleavage prior, log sequence prior, log κ_1 prior, and log κ_2 prior for the peptide $DLVESAPAALK$.

Peptide	**Log Posterior Densities**	**Log Likelihood**	**Log Cleavage Prior**	**Log Sequence Prior**	**Log κ_1 Prior**	**Log κ_2 Prior**
IDVESAPAALK	65.92	103.28	-11.69	-31.12	5.11	0.34
DLVESAPAALK	62.26	100.01	-11.64	-31.26	4.85	0.32
NGGVESAPAALK	61.35	103.55	-12.16	-35.39	5.11	0.23
DLVESAAPALK	54.92	91.64	-10.23	-31.26	4.53	0.24
DIVESAPAALK	49.50	83.52	-7.74	-31.63	5.26	0.10
EPDDSAPAALK	53.16	88.85	-9.39	-31.26	4.73	0.23
ETILSAPAALQ	44.88	85.41	-10.03	-35.31	4.61	0.22
DIVESAPAAIK	44.27	78.78	-7.74	-31.63	4.79	0.07
ETILSAPAALK	42.34	83.38	-14.27	-31.75	4.85	0.12
ETVITAPAALK	39.42	74.75	-8.44	-31.82	4.70	0.23

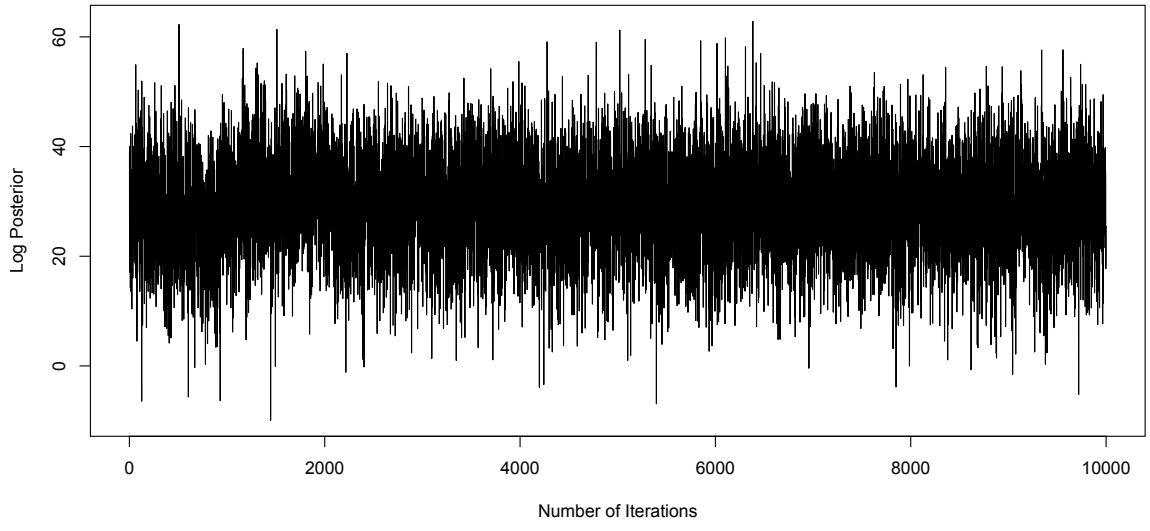

Figure 7.8 Trace plot of the log posterior for the peptide $DLVESAPAALK$.

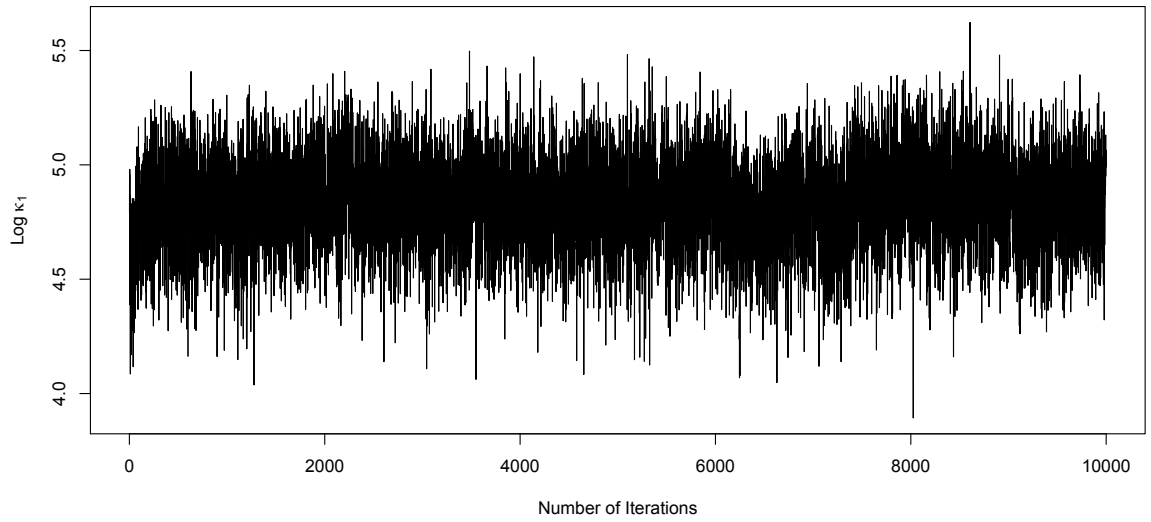

Figure 7.9 Trace plot of log κ_1 for the peptide $DLVESAPAALK$.

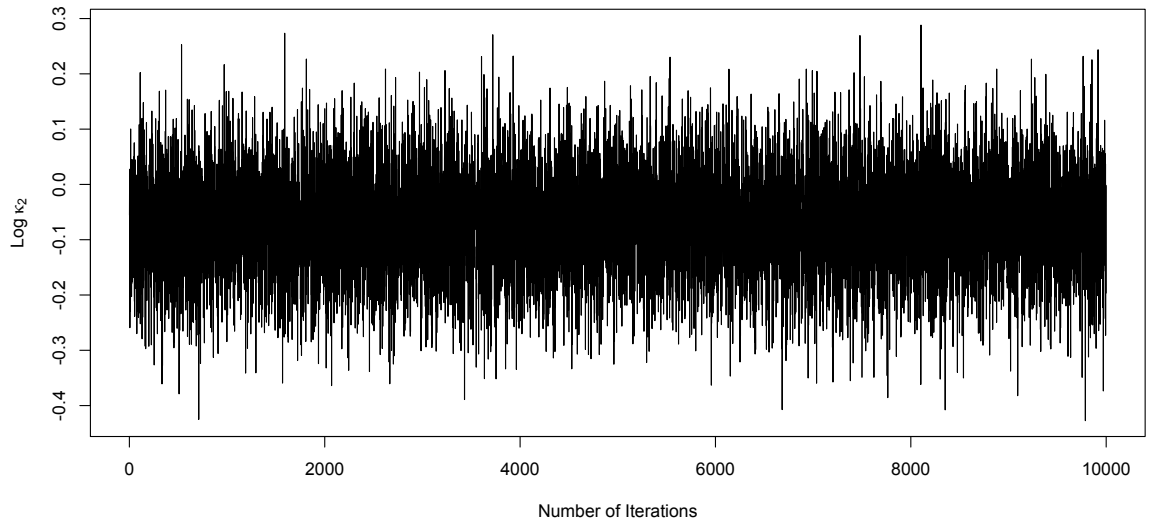

Figure 7.10 Trace plot of log κ_2 for the peptide $DLVESAPAALK$.

We look at results from using different starting peptides that we obtain from PepNovo. Consider the starting peptides: $DLVESAPAPSK$, $LDVTDAPAALK$, and $PMVEGTPAALQ$. Table 7.6 shows the top estimated peptides along with their corresponding log posterior densities for the above starting peptide values. As in Example 1 in Section 7.1, the results are similar and that the true peptide (highlighted in bold in the table) is among the best choices in each case, and the log posterior densities are also similar in the three cases.

7.3 More Examples With Real Peptides

In this section, we look at 10 more peptides whose peptide sequences are of different lengths. Tables 7.7 - 7.16 show the top estimated peptides obtained from our MCMC algorithm along with their corresponding log posterior densities for different peptides. The starting peptide, obtained from PepNovo, is provided for each table, and if the true peptide is in the list, it will be highlighted in bold. In these examples, the true peptide is captured in the list of top estimated peptides in all cases but two.

7.4 Exploring Tuning Parameters

Extensive experimentation showed that using a threshold of 75% works well. We looked at several other threshold values to see which optimized the results. Table 7.17 shows the top estimated peptides along with their corresponding log posterior densities for the peptide $TGMSNVSK$ when using a threshold of 50% and 65% with a starting peptide of $SAMYHSK$, and Table 7.18 shows the top estimated peptides along with their corresponding log posterior densities for the peptide $TGMSNVSK$ when using a threshold of 85% and 95% with a starting peptide of $SAMYHSK$. Table 7.19 shows the top estimated peptides along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ when using a threshold of 50% and 65% with a starting peptide of $DLVESYFLK$, and Table 7.20 shows the top estimated peptides along with their corre-

Table 7.6 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ for three different starting peptides. The true peptide is in bold.

Starting Peptide					
DLVESAPAPSK		**LDVEYYALK**		**PMVEGTPAALQ**	
Peptide	Log Posterior	Peptide	Log Posterior	Peptide	Log Posterior
PMVESAPAALK	54.727117	PMGNGSAAPAIK	57.37	MVVLTAPAAIQ	49.35
MPVESAPAAIK	53.640850	**DLVESAPAALK**	54.21	DLVESAPAAIK	46.97
MPVESAPAALK	50.551788	PMGNGSAPAAIK	50.05	**DLVESAPAALK**	44.64
VEVESAPAALK	49.654627	EVVESAPAAIK	44.81	MPTVDAPAALQ	44.42
MPPSMAPAAIK	40.189160	PMVESAPAAIK	44.78	NNVESAPAAIK	43.35
DLVESAPAALK	38.911162	VEVESAPAALK	44.63	MPTLTAPAALQ	42.34
EVPSMAPAAIK	30.794783	VEVESAPAAIK	43.22	MPVDTAPAALK	41.05
MPPCDAPAAIK	29.576415	DLVESAAPAIK	38.66	MPVESAPAALK	39.07
EVPSMAPAAIQ	18.144605	LDVESAPAAIK	38.58	MVVDVAPAAIQ	36.58
EPVSMAPAALQ	6.496682	LDVESAPAALQ	33.46	MPTDVAPAAIQ	35.52

Table 7.7 The top estimated peptides from the MCMC algorithm for the true peptide $AQLQEIAQTK$ when using the starting peptide $KALQNQAQTQ$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
AQLQANQQTK	31.79
AQLQILSQTK	27.86
AQLQPESQTK	27.85
AQLQLEAQTK	26.09
AQLQEIAQTK	23.26
AQLQKGQQTK	13.21
AQLQKANQTK	9.99
AQLQKGGAQTK	0.58
AQLKANFPIK	-28.36
AVAAQKGGAQTK	-32.17

Table 7.8 The top estimated peptides from the MCMC algorithm for the true peptide $SILSELVR$ when using the starting peptide $AELSGNAVR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
TVLSEIVR	-11.62
TVLSELVGV	-11.80
TVSLELVGV	-12.80
AELSELVR	-15.81
TVLSEIVGV	-16.14
AEISEIVR	-17.69
AELSEIVR	-17.75

Table 7.9 The top estimated peptides from the MCMC algorithm for the true peptide $SVANAEQMDR$ when using the starting peptide $WANAKEMDR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
SVGQAAADMDR	56.02
SVGQAEKMDR	43.50
SRQAEQMDR	43.37
SVGQAEQMDR	30.64
SVANAEQMDR	18.15
SVAGGAEQMDR	-14.84

Table 7.10 The top estimated peptides from the MCMC algorithm for the true peptide $VSEGQTVR$ when using the starting peptide $SVWQSLR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
VSEGGATVR	32.74
VSEGGAVTR	27.25
VSEGGIGTR	25.06
VSSVGAVTR	4.44
VSENAVTR	-12.11
VSEGQISR	-28.94
VSEGQTVR	-31.07
VSEGKVTR	-37.99

Table 7.11 The top estimated peptides from the MCMC algorithm when using the starting peptide $TNVFALPDVVGVLTK$ for the true peptide $AFNEALPLTGVVLTK$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
AFENPSPLTGVVITK	102.16
AFENALPLTGVVITK	67.59
AFNSLLPLTGVVITK	62.71
AFNSILPLTGVVITK	47.81
AFNSILPLTGVVLTK	18.67

Table 7.12 The top estimated peptides from the MCMC algorithm when using the starting peptide $GYAGDGDSDSEVQ$ for the true peptide $GYAGDTATTSEVK$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
YGAGDGDTTSEVK	51.31
GYAGDTATTSEVK	51.00
YGAGDGDAMSEVK	49.10
YGAGDTATMGEVK	44.85
GYAGDTATTSVTR	44.32
YGAGDTATTSEVK	43.29
GYAGDTATTSISR	35.57
GYAGDTATTSNNK	11.65
GYAGDTATTSEVK	-58.71

Table 7.13 The top estimated peptides from the MCMC algorithm for the true peptide $LVSSPSTLNPGTNVAK$ when using the starting peptide $PDSSPSDPDSTLPNR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
LVSSIAITNPASNVAK	29.72
LSVSPSLGGTGVVNVAQ	22.94
LSVSPSLGGTIAAGVQQ	5.63
LSVSPSLGGTLAANVAQ	0.95
LSVSPSLGGTLAAGVQQ	-4.39
LSGEPSLGGTIAAGVQQ	-9.93
LSGEPSLGGTIAAGGVAK	-39.80
LVSSPSLGGTPGTNVAK	-40.27
LSGEPSLGGTIAAGVGAK	-42.24
LVSSPSTLNPGTNVAK	-51.91

Table 7.14 The top estimated peptides from the MCMC algorithm for the true peptide $MPPTEGETGGQVLGSK$ when using the starting peptide $MPPTEGETLEVTRK$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
MPPTEGDDGGQVIGSK	75.73
MPPTEGETGGQVLGSQ	66.21
MPPTEGETGGQVIGSQ	61.55
MPPTEGETGGQVLGSK	50.23
MPPTEGESAGQVLGSK	41.25
MHGTEGESAGQVLGSK	24.38
MHGTEGGDDGQVLGSK	24.24
MHGTEGGMVGQVLGSK	19.38
MHGTEGSTVGQVLGSK	3.63
MHGTEGALCGQVLGSK	-15.43

Table 7.15 The top estimated peptides from the MCMC algorithm when using the starting peptide $GAASDVLSLGGK$ for the true peptide $SGPLAGYPVVDLGVR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
SIPGAGLDDDDLGVR	113.65
SIGPAGIDDDDLGVR	104.58
SIGPAGIVMDDLGVR	93.72
SIGPAGIVFVDLGVR	88.03
SIPGAGIVFVDLGVR	84.06
SIPGAGIMDVDLGVR	76.82
SGPLAGPDFVDLGVR	37.54
SGPLAGIMDVDLGVR	29.99
SGPLAGPYVVDLGVR	4.51
SGPLAGYPVVDLGVR	-10.05

Table 7.16 The top estimated peptides from the MCMC algorithm when using the starting peptide $GHYFEQTWTSQPVK$ for the true peptide $YHFEQSTVTSQPAR$ along with their corresponding log posterior densities.

Peptide	Log Posterior Densities
YHFENTTVTSQPAR	89.49
YHFEQQYPTNPIN	45.71
YHFEQSTVTSQPAR	43.55
YEFHQSTAESPAQR	35.11
YEFHQSTVTSPAQR	27.06
YHFEQQYPTNPAR	16.05
YCYFQSTVTSPAKR	14.25
YHFEQSTCPTNPAR	9.68
HYFEAGSTVTSAPKR	-3.35
YHFEQSTVTTNPAR	-13.56

Table 7.17 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $TGMSNVSK$ for a threshold of both 50% Da and 65% with a starting peptide of $SAMYHSK$. The true peptide is in bold.

Threshold			
50%		**65%**	
Peptide	Log Posterior	Peptide	Log Posterior
TGGATTGMGA	2.56	SSDSNVSK	5.54
TGGATTTSK	1.63	TGMSGGVSK	5.46
TGGATDSSK	1.39	SAMSGGVSK	3.01
TGGATTTSAG	0.74	TGAFSNVK	2.56
TGGATTSASA	-1.34	TGMSSNVK	1.72
TGGASDTSAG	-3.13	TGMSNVSQ	-2.47
TGGATTTGSA	-3.39	TGFANVSK	-2.68
TGGASDTSK	-3.52	TGMSNVSGA	-3.55
GTGATTGMK	-7.33	SSDSNVSQ	-3.94
GTGATASMK	-8.40	**TGMSNVSK**	-5.39

sponding log posterior densities for the peptide $DLVESAPAALK$ when using a threshold of 85% and 95% with a starting peptide of $DLVESYFLK$. One can see that as the threshold is decreased, the estimated peptides become less similar to the true peptide. This happens because as we lower the threshold, more noise enters the observed spectrum and the value of S_2 in the likelihood is greatly increased. Therefore, our algorithm cannot find the true peptide. As the threshold is increased, at a certain point the estimated peptides become less similar to the true peptide. Although there is less noise in the observed spectrum when the threshold is increased, signal peaks may be removed from the observed spectrum with a large threshold. Thus, our algorithm will not be able to correctly to identify the true peptide. Using a threshold of 75% removes many noisy peaks while still retaining the signal peaks.

Although we used a tolerance of 0.5 Da like PepNovo, we explored using a smaller tolerance and larger tolerances. For every peptide in the data set, we calculated the mass for each ion type and found the average percentage of ion presence within a specific tolerance. Table 7.21 gives the average percentage of ion presence for different tolerances for each

Table 7.18 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $TGMSNVSK$ for a threshold of both 85% Da and 95% with a starting peptide of $SAMYHSK$. The true peptide is in bold.

Threshold			
85%		**95%**	
Peptide	Log Posterior	Peptide	Log Posterior
TGSTSGVSQ	-0.33	TGMSASQAA	-1.07
TGMSNVSK	-4.44	TGMSSAQAA	-1.20
TGMSNVSQ	-10.98	TGMSAQSAA	-10.71
TGMGSGVSQ	-13.96	TGMSASAAGA	-25.87
TTGSMNVSK	-19.48	TGMSASAAK	-29.95
TGAFNVSK	-30.62	TGMSATQK	-31.80
SASQSSEK	-34.06	TGMSDGQK	-34.02
SASQTDSK	-35.97	TGSMDGQK	-35.23
SAAFNVSK	-42.30	TGSFVGQK	-36.65
SASMNVSK	-47.57	TGSFFHK	-38.82

Table 7.19 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ for a threshold of both 50% Da and 65% with a starting peptide of $DLVESYFLK$. The true peptide is in bold.

Threshold			
50%		**65%**	
Peptide	Log Posterior	Peptide	Log Posterior
EVVESGTGAHK	113.62	SLGAAGSGVAPVK	62.26
EVVESASAGHK	110.98	AEGAAGSGVAPVK	51.72
EVVESASGAHK	106.83	AEGAAGSINPVK	50.43
EVVESAGSAHK	102.86	AEGAGASIVPNK	50.24
NNVESASAGHK	95.83	SLGAAGSGGIPVK	50.12
IAGTNSTGETY	95.01	LKSQSIYFK	38.91
EVVESASAGHQ	93.12	AEGAGASINPVK	38.65
MPVESTGETY	92.07	AEKQSIYFK	34.88
MPVESASETY	91.56	AEKQSIVPNK	34.42
QLTGGSTGETY	90.07	AEKGASIVPNK	31.50

Table 7.20 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ for a threshold of both 85% Da and 95% with a starting peptide of $DLVESYFLK$. The true peptide is in bold.

Threshold			
85%		**95%**	
Peptide	Log Posterior	Peptide	Log Posterior
AQAKTPQELK	-20.19	IDVDGASQVPI	5.20
AQQATPQELK	-24.62	LDVDGASQVPI	1.13
HSTMHMELK	-35.70	IDVDTNQVPI	0.47
HSTDRQELK	-39.48	ILTDTNQVPI	-11.17
HSCATPQELK	-45.61	ILTDTNINPL	-15.82
HSTVDGQELK	-46.21	ILTDTNINPI	-16.756
HSCLVGQELK	-46.60	ILTDTIMPPL	-17.14
HSCEHMELK	-51.43	ILTDTIMPPI	-27.62
DIVEHMELK	-56.86	LLTDTIMPPI	-37.11
DIVEHLFLK	-72.59	LDVDTIMPPI	-37.30

ion type. One can see that after a tolerance of 0.2 Da, the percentage of ion presence tends to level off for the tolerances. Although the percentage of ion presence is increased when the tolerance is increased, the room for error also increases. There are some other ion types such as $b - H_2O$ and $b - NH_3$ and possibly $y - H_2O, y^2$ and $y - NH_3$ whose presences are detected in the spectrum. Including these ions in the model may provide better identification of the true peptide. Further exploration of other ion types is discussed in Section 8.1.

We then explored how different tolerances affected the choice of the estimated peptides. Table 7.22 shows the top estimated peptides along with their corresponding log posterior densities for the peptide $TGMSNVSK$ when using a tolerance of 0.1 Da and 1.0 Da and with a starting peptide of $SAMYHSK$. Table 7.23 shows the top estimated peptides along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ when using a tolerance of 0.1 Da and 1.0 Da and with a starting peptide of $DLVESYFLK$. Using a small tolerance like 0.1 Da hardly allows for any error in the mass spectrometer. In order for the true peptide to be estimated correctly using a tolerance of 0.1 Da, the observed

spectrum would need to be aligned almost perfectly with the observed spectrum. A large tolerance like 1.0 Da would allow more room for error but it would expand the parameter space that needs to be searched, which could prevent the algorithm from finding the true peptide in an efficient manner.

7.5 Result Comparisons

An important goal is to quantify objectively how our method performs relative to its competitors. Here we compare our results with those of PepNovo. We will use the results from PepNovo using both the rank score and PepNovo score. Since our method hopes to identify peptides that have not been cultured, we do not compare our results to that of SEQUEST and MASCOT. We will not make a distinction between the amino acids I and L because they have identical masses of 113.084. Although PepNovo does not make a distinction between the amino acids K and Q because the difference in their masses is only a minute difference of 0.04 Da, we will make the distinction.

Our comparison method looks at the minimum number of switches in the amino acid sequence of the peptide that it takes to obtain the true peptide. Switches are only considered if the total mass remains within 0.5 Da of the total mass of the true peptide. If the best estimated peptide is the truth, then minimum number of switches would be zero. To illustrate this comparison method, consider the true peptide $VSEGQTVR$ with the best estimated peptide $WEGQTVR$. One can see the only difference from the true peptide is that best estimated peptide begins with W while the true peptide begins with VS. Note that the mass of W (186.079 Da) is within 0.5 Da of the mass of VS (186.1) and thus, the switch can be made. Therefore, by switching W with VS, we obtain the true peptide, and so the minimum number of switches is 1. If more than 3 switches is needed to obtain the true peptide, we denote the minimum number of switches as $3+$. In both the PepNovo rank score and PepNovo score, the best estimated peptide may not have the same mass as the true peptide. PepNovo does provide the $N - Gap$, which is the mass gap from the

Table 7.21 Average percentage of ion presence within a tolerance.

Ion Type	Tol = 0.1	Tol = 0.2	Tol = 0.3	Tol = 0.4	Tol = 0.5	Tol = 0.6	Tol = 0.7	Tol = 0.8	Tol = 0.9	Tol = 1.0
b	29	48.6	57.2	60	61	61.7	62.3	63	64	65
y	33.6	57	65.7	67.3	68	68.4	68.7	69	69.4	70
a	7.6	12	14.4	15.8	16.6	17.3	18	19	20.4	22.4
$b - H_2O$	10.7	20.3	28	32	34.5	36	38	40.5	43.3	45.8
$y - H_2O$	8	13.8	17.2	19.3	20.4	21.4	22.5	24	26.6	30
$b - NH_3$	9.5	19	26.6	30.6	32.4	33.5	34.4	35.6	37.3	40
y^2	8.3	15	18.8	21.8	25.4	29. 4	32. 1	34.2	36.1	36.9
$y - NH_3$	8.6	14.3	17.5	19.2	20.3	21	22	23.3	25.2	28.2
$b - H_2O - H_2O$	4.6	8.4	10.7	12.2	13	14	15.2	16.6	18.8	20.8
$b - H_2O - NH_3$	4.6	8.7	11.4	13.3	14.3	15	16	17	19	21
$a - NH_3$	5.6	9	11.2	12.6	13.5	14	15	16	17.7	20
$a - H_2O$	4.3	7.5	9.3	10.5	11.3	12	13	14.3	16.5	19.3
$y^2 - H_2O$	4.2	8.1	12.1	15. 5	18.1	20.4	23. 2	24.5	26.3	29.2
$y - H_2O - NH_3$	3.8	6.5	8.6	10	11	11.8	12.7	13.9	15.4	17.5
$y - H_2O - H_2O$	3.5	6	7.7	9	10	10.7	11.7	13	14.8	16.6

Table 7.22 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $TGMSNVSK$ for a tolerance of both 0.1 Da and 1.0 Da with a starting peptide of $SAMYHSK$.

Tolerance = 0.1		**Tolerance = 1.0**	
Peptide	Log Posterior	Peptide	Log Posterior
TTSSNSVAG	-1.77	SAMSGGVSGA	0.10
TTSSNSVK	-9.37	SAMSNVSGA	-3.99
TMGSNSVK	-16.78	SAMSGVGSGA	-6.97
MTGSNSVK	-17.12	ASMSNVSGA	-8.12
TGSMNWK	-18.63	ASQFGVSGA	-18.17
TMGSNWK	-18.76	ASQFGVSE	-28.63
TMGSNSLN	-20.53	ASGHVHSE	-41.29
TMGSNSIN	-23.92	ASGHHVSE	-42.70
TGMSNWK	-25.06	SAGHVHSE	-49.90
TMGSNSQV	-29.30	TGGHVHSE	-52.84

Table 7.23 The top estimated peptides from the MCMC algorithm along with their corresponding log posterior densities for the peptide $DLVESAPAALK$ for a tolerance of both 0.1 Da and 1.0 Da with a starting peptide of $DLVESYFLK$.

Tolerance = 0.1		**Tolerance = 1.0**	
MIVLLNLAVK	16.13	EVVESLNPVK	32.61
MIVLLSPNVK	15.13	EVVESYFNK	31.97
MIVIINLAVK	5.20	EVVESLPVNK	28.44
MMHGINLAVK	3.50	EVVESLVPNK	27.67
MMHGGGILAVK	0.78	EVVESYFLE	19.11
MMHGNILAVK	-6.06	EVVESYFIE	18.13
MMHGIVQAVK	-8.46	EVVESYFIQ	0.59
MPVEIVQAVK	-8.89	EVGATSYFIQ	-19.35
MPVEVLQGLK	-20.96	EVGDGSYFIQ	-19.48
MPVEIVQGLK	-25.78	DLGGNSYFLK	-22.42

N-terminal to the start of the de novo sequence and the $C - Gap$, which is the mass gap from the C-terminal to the end of the de novo sequence. While it does provide those mass values, it cannot detect the amino acid residues that should correspond to the mass gaps. For example, consider the true peptide $DLVESAPAALK$ with a total mass of 1113.616 Da. Using the PepNovo rank score, the best estimated peptide is $DNVESLEV$, which has a mass of 885.4088 Da. Note that this mass is the sum of masses of each amino acid residue in the sequence and does not include the mass of a water molecule and hydrogen molecule. That is accounted for in the mass gap. The $C - Gap$ value given is 229.029 Da implying there are amino acid residues missing from the end of the de novo sequence whose mass should total 229.029 Da. We cannot look at the minimum number of switches; however, we do know that a peptide with a total mass less the total mass (outside of the tolerance) of the true protein cannot be the true peptide.

Table 7.24 displays the best estimated peptides for the PepNovo rank score, the PepNovo score, and our method along with the corresponding true peptide. The minimum number of switches is in parentheses. One can see that in most cases when using the PepNovo rank score, the best estimated peptide does not have the correct total mass. An advantage of our method is that all our estimated peptides will have a total mass within a tolerance of the total mass of the true peptide. When comparing the results from the PepNovo score and our method of using the Bayes log posterior, for peptides with shorter amino acid sequences, the PepNovo score tends to do slightly better in estimating the true peptide. However, for peptides with longer amino acid sequences, our method tends to do better in estimating the true peptide. Note that one method does not necessarily work best in every case, yet a combination of two good methods can produce an even better method. Therefore, an avenue to explore in the future is developing a rank score method that will combine our method with PepNovo's method.

Table 7.24 The best estimated peptides using the PepNovo Rank Score, PepNovo Score, and our method (Bayesian posterior) . The last column is the true peptide. The number of switches it takes to obtain the true peptide is in parentheses. (0) denotes the estimated peptide is the true peptide. (*) denotes that the minimum number of switches cannot be found since the estimated peptide does not have the correct total mass.

Best Estimated Peptides			
PepNovo Rank Score	**PepNovo Score**	**Bayes Log Posterior**	**True Peptide**
DNVESLEV (*)	DLVESAPAALK (0)	IDVESAPAALK (1)	DLVESAPAALK
AQLQNQAQTK (1)	VVLQELAQTK (1)	AQLQANQQTK (1)	AQLQEIAQTK
SVANAEQMDR (0)	WANAQEMDR (2)	SVGQAAADMDR (2)	SVANAEQMDR
AELSELV (*)	VTLSELVR (1)	TVLSEIVR (1)	SILSELVR
TGMSNVSK (0)	TGMSNVSK (0)	TGMSGGVSK (1)	TGMSNVSK
VSEGQTVR (0)	VSEGQTVR (0)	VSEGGATVR (1)	VSEGQTVR
QASEVVSLNK (*)	FEHAAASEVVSLGGK (3+)	SIPGAGLDDDDLGVR (3)	SGPLAGYPVVDLGVR
SQESTVTSQPAR (*)	YHFEQESATSQVPK (2)	YHFENTTVTSQPAR (1)	YHFEQSTVTSQPAR
MPPTEGETNQVL (*)	MPPTEGETNQVLGSK (1)	MPPTEGDDGGQVIGSK (1)	MPPTEGETGGQVLGSK
GYAGDTATTSEVK (0)	GYAGDTATTSEVK (0)	YGAGDGDTTSEVK (2)	GYAGDTATTSEVK
NPSSPSDVLSS (*)	VLSSPSDPDSPVQEK (3+)	LVSSIAITNPASNVAK (3)	LVSSPSTLNPGTNVAK
LPDVGVVLTK (*)	NTVFALVLVLAALTK (3+)	AFENPSPLTGVVITK (2)	AFNEALPLTGVVLTK

CHAPTER 8

CONCLUSION

Proteomics produces large amounts of spectra from mass spectrometry. Issues can arise such as post translational modifications (PTMs), mutations, and contaminants causing the spectra to fail to match peptides from a database. Also, there are copious microorganisms such as prokaryotes and eukaryotes that have not been identified and therefore, using a database search to identify these peptides would not prove useful. Of those microorganisms that have been identified, some show evidence of PTMs, which can create complications in the de novo sequencing when comparisons are made between the theoretical spectrum and the observed spectrum. Thus the need for a method of identifying peptides that does not rely on a known database and is not is affected by PTMs is evident.

Protein sequencing is another reason for the need for an accurate peptide identification method. Protein sequencing is the method of identifying the true amino acid sequence of a protein. Identifying an entire protein is almost impossible, and so the protein is split into short peptides. Ergo, being able to correctly identify the amino acid sequence of a peptide will aid in identifying the true protein sequencing.

There are limitations in the current methods for protein identification. Our method hopes to alleviate such drawbacks of de novo sequencing and database searches. By using a Bayesian approach, we allow prior knowledge of the peptides to help us to find the best estimate of the true peptide. In most Bayesian approaches, the posterior distribution can be extremely complicated and thus, Monte Carlo methods are employed. Due to the complexity of our posterior density, we use MCMC simulation to obtain the posterior probabilities. Using such MCMC algorithms allows one to approximate the target distribution, which in

our case is the posterior distribution of the unknown peptide sequence. One advantage of our method is that it is not dependent upon known peptides. We hope that our method will obtain more accurate estimates of the true peptide, helping researchers in the field of proteomic research and potentially aiding in identifying microbes. With the study of proteins becoming more important in identifying early stages of diseases (most commonly cancer), it is of great importance to be able to correctly identify these proteins. The goal is for scientists to use these peptides as biomarkers for diseases. Conceivably, better identification of peptides could aid the diagnosis of types of cancers or find better patient treatments.

8.1 Future Work

It is important to compare our results with other competing peptide identification methods and compare more than just several examples. We plan to compare our results with PEAKS, MASCOT, SEQUEST, and PepNovo using the same comparison technique described in Chapter 7. We plan to look at several hundred peptides and obtain the mean number of minimum switches it takes to obtain the true peptide for each method. When considering peptides of different (and, in particular, longer) amino acid lengths there is good reason to believe our method will compete favorably with others.

Noise peaks are a common problem in peptide identification, and we will explore other thresholding methods. A common preprocessing method is binning, which reduces the amount of data by grouping adjacent m/z values together. Hence it reduces the number of noise peaks in the spectrum. A particular m/z and its corresponding intensity value, within a window, is chosen to represent the group. Selecting the window width can be quite difficult. If the window width is too large, signal peaks may be removed from the dataset, causing the observed spectrum and theoretical spectrum to not be aligned. If the window width is too small, then the purpose of binning is defeated; that is, the number of noise peaks will not be reduced. One bin will be composed of N pairs of m/z and intensity values and will be in the form of $[(I_1, m/z_1), \ldots, (I_N, m/z_N)]$ where I_j is the j th intensity

Original Data		After Binning (Window = 0.5 Da)	
m/z	Intensity	m/z	Intensity
585.9116 586.0613	3 5	585.9865	5
586.3880 586.5022 586.5732	11 1 2	586.4878	11
586.8460	12	586.8460	12
587.4271 587.7023	1387 8	587.5647	1387
587.7939 587.8698 588.0663 588.2092	6 6 5 8	587.9848	8
588.4289 588.6898	520 10	588.5594	520
588.7769 588.9930	6 12	588.8850	12

Figure 8.1 An example of how the data are reduced when using the binning method, obtained from Monroe (2013).

value and m/z_J is the j th m/z value for $j = 1, \ldots, N$. This group of pairs will then be combined into the vector $(I, m/z)$ of pairs. The intensity value I of this bin is found using an aggregate function such as the sum or the maximum of all the N original intensity values, and the m/z value of this bin is found by taking the median or mean of all the N original m/z values (Bachmayer, 2007). Figure 8.1 illustrates how binning reduces the data in a spectrum. Here a window width of 0.5 is used. The maximum intensity value of the group is used to determine the intensity of the bin and mean of the group of m/z values is used to determine the m/z of the bin.

The theoretical spectrum plays an important role in our method and calculating the most accurate spectrum is key. Cleveland and Rose (2012) developed a method to identify better peaks using a neural network, which can be used to construct a predictive model that does not require an extensive understanding of peptide fragmentation. Better identification of peaks will lead to a more accurate theoretical spectrum and ultimately better identification of peptides. Recall from Chapter 3 that other ions can produce peaks in the observed spectrum. Like our method, Cleveland and Rose (2012) concentrates on identifying signal peaks corresponding to b and y ions; they also employ a leveraged neural network (LNN), which is composed of two neural networks used in order to classify peaks. In the first neural network, peak features, such as isotopologues and neutral losses, are found from the data in

Table 8.1 Average number of peaks per spectrum classified as $b-/y-ions$ by LNN and PepNovo, taken from Cleveland and Rose (2012)

Peptide Length	LNN	PepNovo
8	11	43
9	20	46
10	47	51
11	46	52
12	42	58
13	62	55
14	71	77
15	64	87
16	97	72
17	89	107
18	84	89
19	50	93
20	76	93
21	38	102

the spectrum. Then in the second neural network, the results from the first neural network are leveraged as extra features in the second neural network. This process selects peaks with higher precision and reduces the number of peaks in the spectrum, which could make identifying the true peptide more efficient. For additional information about the LNN, see Cleveland and Rose (2012). Table 8.1 shows the average number of peaks selected for each spectrum by LNN and PepNovo. One can see that in almost every case, LNN produced fewer peaks per spectrum.

An important aspect for the future development is to refine the Bayesian model. Clearly the model can be refined by including more signal peaks $b-H_2O, b-NH_3, y-H_2O, y-NH_3, y^2$, isotopic peaks, etc. This will help identify which are signal peaks and which are noise peaks. We will further explore the prior models. Currently the cleavage model is more uniform than that of Huang et al. (2004). A stronger prior could help concentrate the posterior on a more restricted set of candidates.

Adapting the prior to assign probabilities to the length of the peptide is an avenue we wish to explore. That is, we wish to include in the prior a component π_k that measures the

prior probability that the peptide length equals k, for $k = 1, 2, \ldots$. A reasonable choice for this prior would be a Poisson distribution. The current prior favors shorter peptide sequences. In other words, peptides with shorter amino acid sequences have higher prior probabilities than peptides with longer amino acid sequences. Modifying the prior could help alleviate this problem.

We currently are using a fixed dimension parameter space and we will look into using a reversible jump. A reversible jump allows simulation of the posterior distribution on spaces of varying dimensions (Green, 1995). An advantage to using a reversible jump is that it will ensure that our proposed new candidate peptide will have similar posterior strength to our existing candidate peptide. This guarantees that the move and its reverse will both have a good chance of being accepted by the algorithm (Hastie and Green, 2012).

Model selection using the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) will be examined (Akaike, 1974; Gelfand and Dey, 1995). Both criteria are based, in part, on the likelihood function. AIC is a measure of the relative goodness of fit of a statistical model given by $AIC = 2k - 2\ln(L)$ where k is the number of parameters in the model and L is the maximized value of the likelihood function for the estimated model. The model with the smallest AIC value is chosen as the best model. Note the AIC penalizes models with too many parameters, i.e., it discourages overfitting of the model. The BIC is defined as $-2\ln(L) + k\ln(n)$ where k is the number of parameters in the model that need to be estimated, n is the number of observations in the dataset, and L is the maximized value of the likelihood function for the estimated model. The model with the smallest BIC value is chosen as the best model. Like the AIC, the BIC penalizes models with too many parameters, but the penalty term is more severe (Gelfand and Dey, 1995; Burnham and Anderson, 2002; Rodríguez, 2013). With all the variations to our algorithm that we plan to explore, there will be several models defined. Therefore, it will be important to use good criteria to select the best model.

The Bayesian framework and model are in place for practical usage. With some further

modeling refinements, the method we present will be a promising addition to the peptide identification methodological toolbox.

BIBLIOGRAPHY

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43.

Ansong, C., Tolić, N., Purvine, S., Porwollik, S., Jones, M., Yoon, H., P. S., Martin, J., Burnet, M., Monroe, M., Venepally, P., Smith, R., Peterson, S., Heffron, F., McClelland, M., and Adkins, J. (2011). Experimental annotation of post-translational features and translated coding regions in the pathogen salmonella typhimurium. *BMC genomics*, 12(1):433.

Antoniewicz, M. (2013). Tandem mass spectrometry for measuring stable-isotope labeling. *Current Opinion in Biotechnology*, 24(1):48–53.

Bachmayer, S. (2007). Preprocessing of mass spectrometry data in the field of proteomics. Master's thesis, University of Helsinki, Finland.

Burnham, K. and Anderson, D. (2002). *Model selection and multimodel inference*. Springer, Berlin.

Cleveland, J. P. and Rose, J. R. (2012). A neural network approach to the identification of b-/y-ions in MS/MS spectra. *012 IEEE International Conference on Bioinformatics and Biomedicine*, 0:1–5.

Coombes, K. R., Baggerly, K. A., and Morris, J. S. (2007). Pre-processing mass spectrometry data. In Dubitzky, M. Granzow, M. and Berrar, D., editors, *Fundamentals of Data Mining in Genomics and Proteomics*, pages 79–99. Boston: Kluwer.

Damsleth, E. and El-Shaarawi, A. (1989). ARMA models with double-exponentially distributed noise. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(1):61–69.

Dančík, V., Addona, T. A., Clauser, K. R., and Vath, J. E. (1999). De novo peptide sequencing via tandem mass spectrometry: A graph-theoretical approach. In *RECOMB '99: Proceedings of the third annual international conference on Computational molecular biology*, number 135–144, New York, NY, USA. ACM Press.

de Hoffmann, E. (1996). Tandem mass spectrometry: A primer. *Journal of Mass Spectrometry*, 31(2):129–137.

Diamandis, E. P. (2004). Mass spectrometry as a diagnostic and a cancer biomarker discovery tool: Opportunities and potential limitations. *Molecular & Cellular Proteomics*, 3:367–378.

Doob, J. L. (1953). *Stochastic Processes*. John Wiley & Sons, Inc., New York.

Frank, A. (2009). A ranking-based scoring function for peptide-spectrum matches. *Journal of Proteome Research*, 8(5):2241–2252.

Frank, A. and Pevzner, P. (2005). PepNovo: de novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, 77(4):964–973.

Freund, Y., Iyer, R., Schapire, R., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:923–925.

Gelfand, A. and Dey, D. (1995). Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(3):501–514.

Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Hastie, D. and Green, P. (2012). Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Huang, Y., Triscari, J. M., Pasa-Tolic, L., Anderson, A. G., Lipton, M. S., Smith, R. D., and Wysocki, V. H. (2004). Dissociation behavior of doubly-charged tryptic peptides: Correlation of gas-phase cleavage abundance with ramachandran plots. *Journal of American Chemical Society*, 126:3034–3035.

Issaq, H., Conrads, T., Prieto, D., Tirumalai, R., and Veenstra, T. (2003). SELDI-TOF MS for diagnostic proteomics. *Analytical Chemistry*, 75(7):148A–155A.

IUBMB (1992). International Union of Biochemistry and Molecular Biology. In Liébecq, C., editor, *Biochemical nomenclature and related documents*. Portland Press, London. Second Edition.

Kemp, F. (2003). The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. *Journal of the Royal Statistical Society. Series D*, 52(4):698–699.

Lubec, G. and Afjehi-Sadat, L. (2007). Limitations and pitfalls in protein identification by mass spectrometry. *Chemical Reviews*, 107(8):3568–3584.

Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., and Lajoie, G. (2003). PEAKS: Powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 17:2337–2341.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Monroe, M. (Accessed March 2013). Peptide sequence fragmentation modeling. Retrieved from `www.alchemistmatt.com/MwtHelp/PeptideFragModelling.htm`.

Morris, J. S., Baggerly, K. A., Gutstein, H. B., and Coombes, K. R. (2010). Statistical contributions to proteomic research. *Methods in Molecular Biology*, 641:143–166.

Radboud University Nijmegen (Accessed March 2013). MALDI-TOF. Retrieved from http://www.ru.nl/science/gi/facilities/other-devices/maldi-tof/.

Raftery, A. and Akman, V. (1986). Bayesian analysis of a Poisson process with a change-point. *Biometrika*, 73(1):85–89.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag.

Rodríguez (Accessed March 2013). The ABC of model selection: AIC, BIC and the new CIC. Retrieved from http://omega.albany.edu:8008/CIC/me05.pdf.

Rose, J. R., Cleveland, J. P., and Fox, A. (2010). An information theoretic approach to rescoring peptides produced by de novo peptide sequencing. *International Conference on Bioinformatics and Computational Biology (Paris, France), World Academy of Science, Engineering and Technology*, pages 200–205.

SAS Institute Inc. (2009). *SAS/STAT 9.2 User's Guide*. SAS Institute Inc., Cary, NC, second edition.

Schulze, W. (2004). Environmental proteomics - what proteins from soil and surface water can tell us: a perspective. *Biogeosciences Discussions*, 1:195–218.

Singla, P. and Domingos, P. (2005). Discriminative training of Markov logic networks. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 868–873.

Sorensen, D. and Gianola, D. (2002). *Likelihood, Bayesian and MCMC methods in quantitative genetics*. Springer, New York.

Standardbase-Techniques (Accessed June 2012). High performance liquid chromatography. Retrieved from www.standardbase.com/tech/HPLC.pdf.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728.

Visintin, I., Feng, Z., Longton, G., Ward, D. C., Alvero, A. B., Lai, Y., Tenthorey, J., Leiser, A., Flores-Saaib, R., Yu, H., Azori, M., Rutherford, T., Schwartz, P. E., and Mor, G.

(2008). Diagnostic markers for early detection of ovarian cancer. *Clinical Cancer Research*, 14:1065–1072.

Wulfkuhle, J. D., Liotta, L. A., and Petricoin, E. F. (2003). Early detection: Proteomic applications for the early detection of cancer. *Nature Reviews Cancer*, 3:267–275.

Wysocki, V. H., Cheng, G., Zhang, Q., Herrmann, K. A., Beardsley, R. L., and Hilderbrand, A. E. (2006). Peptide fragmentation overview. In Lifshitz, C. and Laskin, J., editors, *Principles of Mass Spectrometry Applied to Biomolecules*, pages 277–300. Wiley-Interscience.

Xu, C. and Ma, B. (2006). Software for computational peptide identification from MS-MS data. *Drug Discovery Today*, 11(13-14):595–600.

Yates, J. R., Ruse, C. I., and Nakorchevsky, A. (2009). Proteomics by mass spectrometry: approaches, advances, and applications. *Annual Review of Biomedical Engineering*, 11:49–79.